

## АНАЛІЗ ПІДХОДІВ ДОСТУПУ ДО ДАНИХ У МУЛЬТИ-КЛАУД СЕРЕДОВИЩІ

**Касерес А.** – аспірант Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», кафедра інформаційних технологій в телекомунікаціях, Київ, Україна.

**Глоба Л. С.** – професор, д-р техн. наук, кафедра інформаційних технологій в телекомунікаціях інституту телекомунікаційних систем національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

### АНОТАЦІЯ

**Актуальність.** Мультихмарна система характеризується послідовним або одночасним використанням послуг від різних хмарних постачальників для виконання програм. Така система є бажаною інфраструктурою для переважної більшості ІТ-бізнесу сьогодні. Наразі існують різноманітні підходи для об'єднання хмарних платформ кількох постачальників. У даній статті досліджувались практичні підходи для досягнення мультихмарної сумісності, зосереджуючись на абстрактному доступі до даних між різними постачальниками cloud-сховищ та multi-cloud розподілі обчислювальних ресурсів. Представлено ключові технології та методології безперервного керування даними, такі як використання мультихмарних шлюзів зберігання даних (на прикладі S3Proху), впровадження платформ керування даними (Apache NiFi) та використання універсальних хмарних бібліотек (Apache Libcloud). У роботі висвітлено переваги і недоліки обраних підходів і проведено експерименти по визначенню вартості і продуктивності для них. Результатом проведених досліджень є визначення вартості і продуктивності для різних підходів доступу до даних у мультихмарних середовищах.

**Мета.** Дослідити різні підходи мультихмарного доступу до даних і визначити найбільш оптимальний за характеристиками продуктивності і вартості.

**Метод.** Запропоновано оптимізацію мультихмарних інфраструктур на основі експериментальних даних. Експериментальне моделювання включає в себе емпіричні вимірювання швидкодії і порівняння витрат на зберігання. Визначення продуктивності базується на вимірюванні часу читання даних і затримки. Для оцінки вартості використовується модель ціноутворення AWS S3. Описано підходи оптимізації з урахуванням розмірів файлів і обсягом зберігання даних: об'єднання сильних сторін різних мультихмарних підходів і динамічне перемикання між рішеннями. Запропоновано алгоритм вибору мультихмарних підходів, де враховані критерії вартості і швидкодії, а також їх пріоритетності.

**Результати.** У ході експерименту отримано значення вартості, необхідної для зберігання і завантаження даних різних обсягів (100 ГБ, 1 ТБ, 10 ТБ), а також продуктивності для передавання файлів різного розміру (100 КБ, 1 МБ, 10 МБ) для технологій мультихмарних шлюзів, платформ управління даними, та хмарно-незалежних бібліотек. S3Proху має найшвидший доступ до файлів для великих обсягів даних. Apache Libcloud показує кращу вартість на невеликих об'ємах. Проте ці обидва підходи значно переважають Apache NiFi. Дане дослідження може сприяти розвитку методів ефективного керування ресурсами у мультихмарних середовищах.

**Висновки.** Отримані результати дають можливість визначити пріоритетність вибору означених парадигм, щоб допомогти організаціям розробити та розгорнути ефективні мультихмарні стратегії, які дозволять їм скористатися перевагами унікальних функцій кожного хмарного провайдера, зберігаючи при цьому уніфіковане, гнучке та ефективне середовище зберігання та обчислення.

**КЛЮЧОВІ СЛОВА:** хмарні обчислення, мультихмарний доступ, універсальні хмарні бібліотеки.

### АБРЕВІАТУРИ

DMP – Data management platform.

### НОМЕНКЛАТУРА

$A$  – множина мультихмарних підходів;

$A_{opt}$  – оптимальний мультихмарний підхід;

$C_i$  – вартість використання хмарного сервісу  $i$ -го провайдера;

$C_{min}$  – мінімальна вартість використання хмарного сервісу;

$CC$  – критерій вартості;

$CT$  – критерій часу доступу до файлу;

$FS$  – розмір файлу;

$K$  – кількість цінових факторів;

$M$  – кількість передавань файлу;

$N$  – кількість досліджуваних мультихмарних стратегій;

$P_i$  – продуктивність хмарного сервісу  $i$ -го провайдера;

$P_{max}$  – максимальна продуктивність хмарного сервісу;

$T$  – середній час доступу до файлу;

$V$  – об'єм зберігання.

### ВСТУП

Проблему об'єднання хмарних платформ кількох вендорів (постачальників послуг) у вигляді єдиного простору зберігання та обробки даних можна зустріти у різних дослідженнях [1–3]. Глобальне впровадження мультихмарних стратегій зростає, і організації всіх розмірів та галузей все більше використовують переваги декількох хмарних провайдерів. Опитування Gartner [4] показує, що 81% користувачів публічних хмар працюють з двома чи більше cloud-провайдерами. Flexera 2024 State of the Cloud Report [5] підтверджує це, показуючи, що 89% усіх організацій використовують мультихмарні технології (рис. 1).



Рисунок 1 – Стан використання хмарних технологій

Оскільки все більше компаній приймають цю тенденцію, проблеми, пов'язані з керуванням доступом до даних у різних постачальників хмарних сховищ, стають дедалі очевиднішими. Це потребує повного розуміння різноманітних підходів, які використовують для абстрактного доступу до даних у мультихмарних середовищах. Зазвичай величезні обсяги даних зберігаються в одній хмарі, а їх обробка відбувається в іншій, а результати обробки можуть бути збережені ще на одній cloud-платформі [3].

Управління доступом до даних у різних постачальників хмарних сховищ і розподіл обчислювальних ресурсів між різними хмарами може становити значні труднощі. Через відсутність стандартних інтерфейсів для досягнення мультихмарної сумісності нерідко інтеграцію потрібно щоразу розробляти вручну [6]. Щоб використати потенціал мультихмарних сховищ і обчислень, організації повинні визначити найкращий підхід до абстрактного доступу до даних у різних хмарних провайдерів, зберігаючи при цьому уніфіковане та ефективне середовище.

**Об'єкт дослідження** – мультихмарні стратегії та їхнє глобальне впровадження.

**Предмет дослідження** – методи та проблеми управління доступом до даних у мультихмарних середовищах, а також порівняння різних підходів до абстрактного доступу до даних з точки зору вартості та продуктивності.

© Касерес А., Глоба Л. С., 2025  
DOI 10.15588/1607-3274-2025-1-16

**Метою** проведених досліджень є постановка та проведення експерименту, який порівнює різні підходи мультихмарного доступу до даних на основі вартості і продуктивності. Оцінюючи та ранжуючи ці підходи, організації можуть приймати обґрунтовані рішення, які оптимізують їх мультихмарне розгортання та максимізують переваги, отримані від використання кількох хмарних постачальників.

На сьогоднішній день широко застосовують такі технології доступу до даних у мультихмарних середовищах, використовуючи верхні рівні абстракції:

- Multi-cloud шлюзи (Multi-cloud storage gateways);
- платформи керування даними (Data management platforms або DMPs);
- брокери хмарних сервісів (Cloud Service Brokers);
- універсальні хмарні бібліотеки (Cloud-agnostic libraries)
- контейнеризація (Containerization);
- оркестрація контейнерів (Container Orchestration).

## 1 ПОСТАНОВКА ЗАДАЧІ

Для визначення ефективності технології доступу до даних у мультихмарних середовищах, використовуючи верхні рівні абстракції моделі OSI, опишемо вхідні і вихідні параметри.

**Дано:** у якості вхідних параметрів виступають підходи абстрактного доступу до даних у мультихмарних середовищах  $A = \{A_1, A_2, \dots, A_N\}$ .

Для оцінки та порівняння різних підходів будуть використані наступні критерії:

- вартість  $C_i$  використання хмарного сервісу  $i$ -го провайдера, де  $1 \leq i \leq N$ .
- продуктивність (час відгуку)  $P_i$  хмарного сервісу  $i$ -го провайдера, де  $1 \leq i \leq N$ .

Критерії та обмеження оцінки якості результатів:

- для вартості важлива мінімізація значення  $C \rightarrow \min$ , при цьому  $C = \{C_1, C_2, \dots, C_N\}$ ;
- для продуктивності важлива максимізація значення  $P \rightarrow \max$ ,  $P = \{P_1, P_2, \dots, P_N\}$ .

**Знайти:** дане дослідження має на меті отримати мінімальну вартість і максимальну продуктивність.

Продуктивність визначимо як відношення обсягу даних до часу обробки (1):

$$P = \frac{FS}{T}. \quad (1)$$

Вартість визначається як сума складових вартості (2):

$$C = \sum_i C_i. \quad (2)$$

## 2 ОГЛЯД ЛІТЕРАТУРИ

Існує багато способів використання ресурсів різних хмар. У [7–9] наведено огляд мультихмарних технологій та прототипів, серед яких використання мультихмарних платформ, шлюзів, прототипи для комплексної обробки подій в хмарному середовищі, розробки для обробки комерційних хмарних симуляцій, фреймворки, відкриті бібліотеки. Найбільш ефективним методом для забезпечення взаємодії є надання гнучких стандартів для роботи у хмарних сервісах.

Зусилля зі стандартизації мультихмарного доступу стосуються ініціатив галузевих консорціумів, організацій зі стандартизації та відкритих спільнот для створення спільного набору специфікацій і рекомендацій, які сприяють сумісності доступу до даних у кількох хмарах.

Деякі зусилля зі стандартизації в мультихмарному просторі доступу до даних включають:

- Інтерфейс керування хмарними даними (CDMI) [10], розроблений Асоціацією індустрії мереж зберігання даних (SNIA), є стандартом, який визначає функціональний інтерфейс для доступу та керування даними, що зберігаються в хмарі. CDMI дозволяє користувачам створювати, отримувати, оновлювати та видаляти елементи даних із хмари, а також виявляти базову інфраструктуру хмарного зберігання й керувати нею.

Позитивним стороною даної технології можна назвати відсутність прив'язки до одного хмарного постачальника, що дає можливість переміщувати дані між хмарами, також являється відкритим стандартом і підтримується багатьма провайдерами.

Разом з тим, стандарт може негативно вплинути на продуктивність в окремих випадках, а також вимагати додаткових зусиль інтеграції з існуючими системами управління даними.

- Відкритий інтерфейс хмарних обчислень (OCCI) [11] – це набір специфікацій, які визначають API загального призначення для керування різноманітними ресурсами хмарних обчислень, включаючи служби зберігання, обчислення та мережеві служби. OCCI прагне сприяти взаємодії між різними хмарними провайдерами та сприяти розробці портативних хмарних програм.

Цей інтерфейс спрощує інтеграцію та управління хмарними ресурсами, може адаптуватися до різних потреб користувачів і технологій.

Водночас, за рахунок модульної структури може призвести до складності у виборі конфігурації під конкретні потреби.

Таким чином, описані стандарти мають деякі спільні переваги, такі як спрощення інтеграції та управління хмарними ресурсами, дозволяє використовувати різні хмарні платформи без прив'язки до одного вендора. Основні недоліки визначаються як потенційні проблеми з продуктивністю і складність впровадження.

Хоча стандартизацію часто вважають найбільш перспективним способом забезпечення взаємодії з кількома хмарами, впровадження стандартів відбува-

ється повільно, оскільки постачальники хмарних послуг зацікавлені в просуванні своїх власних API і технологій.

Таким чином, відсутність загально визначених стандартів вимагає дослідження альтернативних рішень щодо інтеперабельності [9].

Для подальшого дослідження будуть розглянуті наступні парадигми мультихмарного доступу до даних:

- multi-cloud шлюзи;
- платформи керування даними;
- хмарно-незалежні бібліотеки.

1. Multi-cloud шлюзи – це апаратне або програмне забезпечення, яке надає плавний доступ до даних, що зберігаються в кількох хмарних провайдерах, діючи як посередник між програмами користувача та службами хмарного зберігання [12].

Мультихмарні шлюзи сховищ працюють, перетворюючи API-інтерфейси різних хмарних провайдерів в єдиний API, полегшуючи переміщення даних і управління ними. Вони дозволяють користувачам отримувати доступ і керувати даними в різних хмарних провайдерах без необхідності впровадження та підтримки кількох інтеграцій API. Спрощена архітектура доступу до даних через шлюз зберігання проілюстрована на рис. 2.

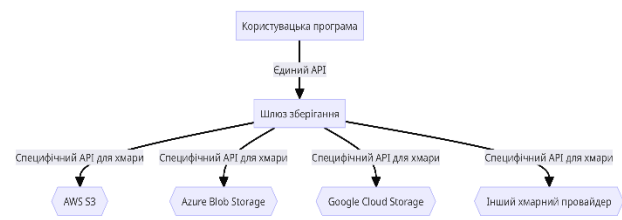


Рисунок 1 – Спрощена схема шлюзу мультихмарного сховища

Доступ до хмарного сховища через шлюз зазвичай має такі переваги:

- Спрощене керування даними за допомогою єдиного уніфікованого API.
- Унікає блокування постачальника та забезпечує стійкість даних шляхом розподілу даних між кількома хмарними провайдерами.
- Потенційно покращена продуктивність доступу до даних за допомогою кешування та методів оптимізації.

При цьому до недоліків можна віднести:

- Додаткові витрати за користування шлюзом.
- Зниження продуктивності або збільшення затримки через додатковий рівень абстракції.
- Залежність від постачальника шлюзу щодо оновлень, виправлення помилок і підтримки.

Розробками у цій сфері є: Cloudbian Hyper-Store, Nasuni Cloud File Services, Panzura Freedom, Morro Data CloudNAS, S3Proху, які оптимізують використання сховища та підвищують безпеку, одночасно забезпечуючи мультихмарний доступ до даних.

2. Платформи керування даними – це програмні рішення, які, на відміну від шлюзів зберігання, призначені для полегшення не лише керування даними та доступу до них у мультимедійних середовищах, але й міграції, захисту та керування даними між різними постачальниками хмарних сховищ. Ці платформи абстрагують базові служби хмарного зберігання та забезпечують централізовану площину

управління даними, що дозволяє організаціям запроваджувати політики, автоматизувати процеси та отримувати аналіз своїх даних у кількох хмарах [13].

Архітектура та потік доступу DMP дуже схожі на шлюзи зберігання (рис. 2). Замість цього пропонується зосередитися на ключових відмінностях для кінцевого користувача.

Діаграма послідовності дій показана на рис. 3.

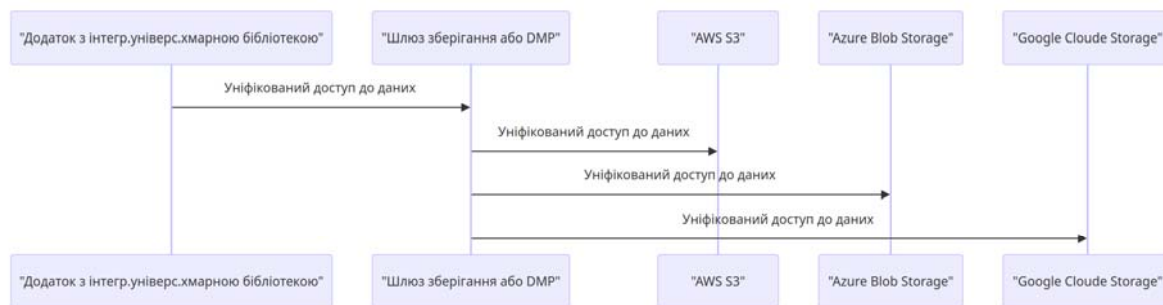


Рисунок 3 – Діаграма послідовності дій щодо інтеграції хмарно-незалежної бібліотеки в кінцевий додаток

Переваги платформ управління даними:

- Оптимізують керування даними, обробку та аналітику в багатьох хмарних провайдерах, пропонуючи більш розширені функції, ніж шлюзи зберігання.
- Покращують управління даними та відповідність вимогам, забезпечуючи централізований контроль і видимість даних із різних джерел, у тому числі мультимедійних служб зберігання.
- Підтримують розширену обробку даних і можливість аналітики, які зазвичай недоступні в шлюзах зберігання.

Недоліки платформ управління даними:

- Можуть бути складнішими у впровадженні та обслуговуванні завдяки додатковим функціям і функціям, які надає платформа.
- Можуть призвести до підвищення вартості використання платформи.
- Деякі платформи можуть не підтримувати всіх хмарних постачальників або певні служби зберігання, що може призвести до потенційних проблем із сумісністю або обмежень.

Серед популярних прикладів DMPs можна назвати Talend Data Fabric, Informatica Intelligent Cloud Services, Dell Boomi AtomSphere, SnapLogic Intelligent Integration Platform, а також рішення з відкритим кодом Apache Nifi.

3. Універсальні хмарні бібліотеки – це програмні інструменти, які забезпечують єдиний уніфікований інтерфейс для взаємодії з декількома службами хмарного сховища на рівні програмування додатків, усуваючи необхідність працювати з кількома спеціальними API для cloud-середовища. Ці бібліотеки часто розробляються та підтримуються спільнотою учасників, що робить їх ініціативами з відкритим вихідним кодом і дозволяє будь-яким розробникам додавати відсутні функції та налаштовувати інструмент відповідно до потреб власної програми. Однак, оскільки ці

бібліотеки потрібно впроваджувати в кінцеву програму на набагато нижчому рівні, вони можуть потребувати більше зусиль у розробці порівняно з попередніми двома рішеннями.

Приклади універсальних хмарних бібліотек і фреймворків включають Apache Libcloud і JClouds. Вони пропонують розробникам можливість взаємодії з широким колом постачальників хмарних сховищ через єдиний API. Хоча обидві бібліотеки схожі за своєю здатністю надавати інтерфейси, що не залежать від хмари, вони відрізняються підходом до інтеграції хмарного сховища [14].

Використання універсальних хмарних бібліотек додає значних зусиль для реалізації кінцевого додатку. Абстрагуючи деталі доступу до даних, специфічні для хмари, це вводить потребу в реалізації власного спеціального API, який зазвичай більш обмежений і менш документований, ніж API хмарних провайдерів. Проілюструємо процес інтеграції хмарно-незалежної бібліотеки за допомогою діаграм 3 і 4.

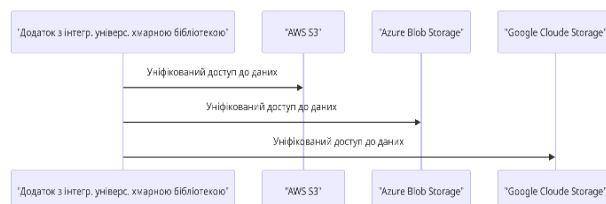


Рисунок 4 – Діаграма послідовності дій щодо інтеграції хмарно-незалежної бібліотеки в кінцевий додаток

Переваги універсальних хмарних бібліотек включають:

- Рентабельність, оскільки більшість рішень з відкритим кодом можна використовувати безкоштовно.
- Розвинуті засоби налаштування, що дозволяють необмежену адаптацію або розширення.

– Швидше виправлення помилок завдяки розробці, керованій спільнотою.

В якості недоліків можна відзначити:

– Зазвичай вимагає більше технічних знань і зусиль для впровадження та підтримки.

– Підтримка та документація можуть бути обмежені, залежно від популярності проєкту та участі спільноти.

– Потенційні проблеми сумісності або обмеження певних хмарних постачальників або служб зберігання [15].

Для аналізу кожного з описаних вище підходів, необхідно провести експериментальні дослідження, у ході яких буде отримано значення показників вартості і продуктивності для подальшої оцінки ефективності впровадження і доцільності застосування кожного зі способів в конкретних умовах.

Для проведення експерименту обрано рішення, які є репрезентативними для кожної парадигми. Визначені технології характеризуються своєю доступністю, тобто мають ліцензію для некомерційного використання, а також стабільну підтримку:

– S3Proху [16] – репрезентує мультихмарні шлюзи. Це серверний компонент, який служить як посередник між додатком та об'єктом сховища.

– Apache NiFi [17] – представник платформ управління даними, розроблений Apache Software Foundation. Це система для візуального програмування потоків даних між різними системами, яка дозволяє легко налаштовувати, керувати та проводити моніторинг потоків даних, включаючи інтеграцію з різними джерелами даних, їх обробку та передачу в різні системи або сховища.

– Apache Libcloud [18] – представник хмарних бібліотек та ініціатив з відкритим кодом. Це бібліотека Python, яка пропонує уніфікований API для взаємодії з широким колом хмарних провайдерів. Забезпечує високий рівень абстракції, який спрощує роботу з різними службами хмарного сховища, що робить його придатним для розробників, які віддають перевагу одній мові програмування та потребують широкого спектру підтримки постачальників.

### 3 МАТЕРІАЛИ ТА МЕТОДИ

У експерименті буде виконано порівняння параметрів вартості  $C$  і продуктивності  $P$  [19].

Параметр вартості в даному експерименті спрямований на оцінку фінансових наслідків, пов'язаних із впровадженням і експлуатацією кожного рішення – S3Proху, Apache NiFi та Apache Libcloud. Усі розрахунки вартості в цьому експерименті базуються на моделях ціноутворення AWS S3 [20] станом на 1 червня 2024 року.

Розглянемо прямі витрати, пов'язані з функціонуванням цих рішень, дозволяючи користувачам приймати обґрунтоване рішення з урахуванням їхніх конкретних моделей використання та бюджетних обмежень.

Аналіз продуктивності має на меті оцінити можливість доступу до даних, які пропонує кожне рішення, і зосереджений на вимірюванні часу читання даних і затримки.

Для проведення цього експерименту зосереджуємось на таких складових вартості:

– Витрати на зберігання. Розраховуються на основі обсягу даних, що зберігаються в хмарі. Ми розглядаємо різні обсяги даних (100 ГБ, 1 ТБ і 10 ТБ).

– Витрати на екземпляр (віртуальна машина). Ці витрати пов'язані із запуском екземплярів віртуальних машин на хмарних платформах, зокрема на AWS EC2 для даного експерименту. Розглядаємо розміри екземплярів віртуальних машин, які можуть обробляти обсяги даних, що використовуються в тестах.

– Витрати на запит. Такі витрати оцінюють фінансовий вплив кількості запитів PUT, GET і LIST, зроблених під час операцій доступу до даних. Ці витрати зазвичай залежать від кількості запитів, зроблених на місяць.

Усі складові вартості були взяті з офіційного сайту провайдера AWS [20].

Таким чином, загальна вартість для  $i$ -го підходу буде розрахована за (3):

$$C_i = \sum_{j=1}^k C_j. \quad (3)$$

Оскільки всі рішення, з якими проводяться експерименти, є відкритими, ліцензійні витрати не стягуються.

Щоб оцінити продуктивність, файли різних розмірів використано для перевірки часу доступу:

- 100 КБ;
- 1 МБ;
- 10 МБ.

Кожен файл отримано 20 разів для кожного сценарію, щоб отримати найбільш достовірне середнє значення. Для узгодженості використано стабільне кабелне з'єднання зі швидкістю 100 Мбіт/с.

Середній час доступу до файлу для  $i$ -го підходу буде розраховано за (4):

$$T_i = \frac{\sum_{j=1}^M T_j}{M}. \quad (4)$$

Наступні кроки описують процес вимірювання продуктивності:

– Провести порівняльний аналіз для кожного рішення, щоб виміряти час, необхідний для читання даних від різних постачальників хмарних сховищ.

– Оцінити затримку для кожного рішення, визначивши час, витрачений на обробку та повернення запитів.

– Проаналізувати пропускну здатність кожного рішення, вимірюючи обсяг даних, до яких можна отримати доступ і передати їх за одиницю часу.

Продуктивність у нашому випадку – це швидкість обробки даних, тобто, чим менше значення середнього часу доступу до файлу, тим краще.

Таким чином, після проведення вимірювань для кожного підходу будуть визначені вартість і швидкість доступу до файлів.

#### 4 ЕКСПЕРИМЕНТИ

Для проведення експериментів програми було встановлено у ізолюванні докер-контейнери із наступним обмеженням ресурсів:

- 2 Virtual CPU;
- 2 GB RAM;
- 2 GB SWAP;
- 16 GB Virtual disk.

##### 1. Експеримент S3Proху

Для проведення експерименту з S3 Proху було використано офіційний Docker-образ [21]. Для вимірювання продуктивності було написано прикладну програму на Python, доступну у відкритому доступі [22].

Експеримент, проведений із застосуванням технології S3Proху, дає уявлення про його можливості як мультимарного рішення для доступу до даних (табл. 1).

Таблиця 1 – Швидкість доступу до файлів S3Proху

Розмір файлу	Середній час доступу, с
100 КБ	0,0732
1 МБ	0,1055
10 МБ	0,3652

Продуктивність S3Proху оцінювалася шляхом вимірювання середнього часу доступу до файлів різного розміру. При стабільному кабельному з'єднанні на 100 Мбіт/с кожен файл було завантажено 20 разів та визначено середнє значення.

Виявлено, що середній час доступу до файлу розміром 100 КБ становить 0,0732 секунди. Середній час доступу дещо збільшився до 0,1055 секунди для файлу розміром 1 МБ і суттєво збільшився до 0,3652 секунди для файлу розміром 10 МБ.

У нашому дослідженні рішення S3Proху фактор вартості оцінювався на основі компонентів, поданих раніше: витрати на зберігання, витрати на екземпляри та витрати на запити. У таблиці 2 подано детальний розподіл цих витрат для даних розміром 100 ГБ, 1 ТБ і 10 ТБ.

Таблиця 2 – Вартість S3Proху

Об'єм даних	Вартість зберігання, USD	Витрати на екземпляр віртуальної машини, USD	Вартість запити, USD
100 ГБ	2,45	7,2	7,27
1 ТБ	24,5	14,4	89,25
10 ТБ	245	72	889,18

##### 2. Експеримент Apache NiFi

Для даного підходу використано Docker-образ [23]. Код програми, який реалізує вимірювання продуктивності, знаходиться у відкритому доступі [24].

Результати тестування Apache NiFi наведено в таблиці 3.

Таблиця 3 – Швидкість доступу до файлів Apache NiFi

Розмір файлу	Середній час доступу, с
100 КБ	7,2302
1 МБ	7,2502
10 МБ	7,2402

Рішення Apache NiFi також було проаналізовано на основі середнього часу доступу до файлів різного розміру. Середній час доступу був значно довшим порівняно з S3Proху: файл розміром 100 КБ займав у середньому 7,2302 секунди. Час доступу для більших файлів, 1 МБ і 10 МБ, був подібним і становив 7,2502 секунди і 7,2402 секунди відповідно.

Для рішення Apache NiFi оцінка вартості відбувається за тією ж схемою, що й аналіз S3Proху. Вартість реалізації Apache NiFi представлена у таблиці 4.

Таблиця 4 – Вартість Apache NiFi

Об'єм даних	Вартість зберігання, USD	Витрати на екземпляр, USD	Вартість запити, USD
100 ГБ	2,45	12,6	7,28
1 ТБ	24,5	25,2	89,27
10 ТБ	245	126	867,88

##### 3. Експеримент Apache Libcloud

Так як Apache Libcloud – бібліотека, що інтегрується прямо у код кінцевого застосунку, даний підхід не потребує проміжного кроку із запитом в окремий контейнер. Відповідно, код бібліотеки було взято з її офіційного репозиторію [25] і інтегровано згідно з наявними інструкціями. Результуючий код експерименту можна побачити по посиланню [26].

Експеримент із Apache Libcloud, бібліотекою Python, створеною для взаємодії з багатьма популярними постачальниками хмарних послуг за допомогою єдиного API, показав наступні результати (табл. 5):

Таблиця 5 – Швидкість доступу до файлів Apache Libcloud

Розмір файлу	Середній час доступу, с
100 КБ	0,0919
1 МБ	0,1059
10 МБ	0,2927

У випадку рішення Apache Libcloud спостерігалося, що середній час доступу до файлів різного розміру був трохи довшим, ніж для S3Proху, але значно коротшим, ніж для Apache NiFi. У таблиці 6 наведено розподіл витрат на впровадження Apache Libcloud.

Таблиця 6 – Вартість Apache NiFi

Об'єм даних	Вартість зберігання, USD	Витрати на екземпляр, USD	Вартість запиту, USD
100 ГБ	2,45	0	9,06
1 ТБ	24,5	0	92,84
10 ТБ	245	0	928,48

Розглянувши кожне рішення окремо, важливо провести порівняльний аналіз між трьома підходами: S3Proху, Apache NiFi та Apache Libcloud. Цей порівняльний огляд дозволяє більш повно зрозуміти сильні та слабкі сторони кожного підходу по відношенню один до одного. Загальна вартість по кожному підходу представлена у таблиці 7.

Таблиця 7 – Загальна вартість

Підхід	Об'єм	Загальна вартість, USD
S3Proху	100 ГБ	16,92
	1 ТБ	128,15
	10 ТБ	1206,18
Apache NiFi	100 ГБ	22,33
	1 ТБ	138,97
	10 ТБ	1238,88
Apache Libcloud	100 ГБ	11,51
	1 ТБ	117,34
	10 ТБ	1173,48

Загальна швидкість доступу до файлів по кожному підходу представлена у таблиці 8.

Таблиця 8 – Швидкість доступу до файлів

Підхід	Об'єм	Середній час доступу, с
S3Proху	100 КБ	0,0732
	1 МБ	0,1055
	10 МБ	0,3652
Apache NiFi	100 КБ	7,2302
	1 МБ	7,2502
	10 МБ	7,2402
Apache Libcloud	100 КБ	0,0919
	1 МБ	0,1059
	10 МБ	0,2927

## 5 РЕЗУЛЬТАТИ

В ході експерименту були отримані дані по вартості і швидкості передавання (тобто час, за який передається файл певного розміру) для кожного з підходів мультимарного доступу до даних. У випадку критерію продуктивності, перевагу варто надати тому підходу, який має найменший середній час доступу, для вартості – найнижчу ціну. Аналізуючи таблиці з відповідними даними можна зробити висновок, що Apache NiFi має найвищу ціну і найбільший час доступу, а, значить, серед обраних підходів, має найнижчий пріоритет вибору, якщо мова йде про вартість і швидкодію. Найкращі ж результати показало використання Apache Libcloud у вартісному показнику. Що стосується продуктивності, то і Apache Libcloud, і S3Proху показали досить близькі результати, проте на розмірі файлів до 1 МБ трохи виграє S3Proху.

Графіки загальної вартості і швидкості доступу до файлів показані відповідно на рис. 5 і 6.

Тепер можна розглянути підходи для оптимізації.

Об'єднання сильних сторін:

– S3Proху має найшвидший доступ до файлів розміром до 1 МБ;

– Apache Libcloud показує найкращу вартість на різних обсягах.

– Apache NiFi підходить для сценаріїв, де швидкодія не є критичною, проте необхідна складна оркестрація даних.

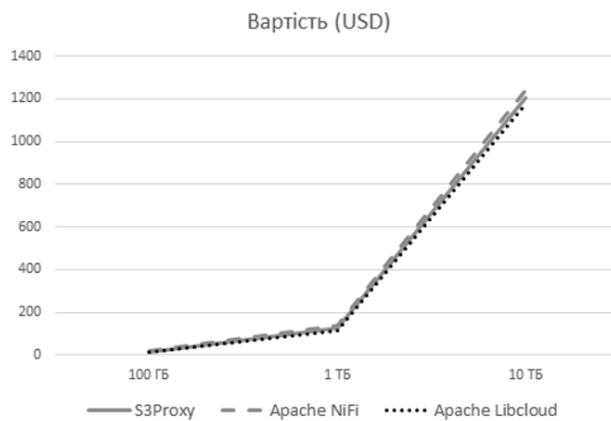


Рисунок 5 – Графік залежності вартості від обсягу зберігання

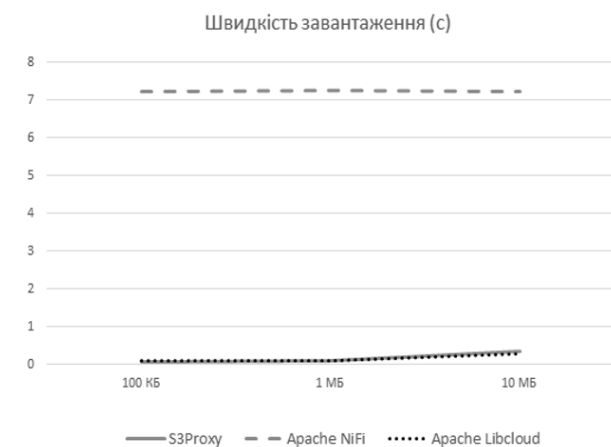


Рисунок 6 – Графік залежності часу передачі даних від розміру файлу

Динамічне перемикання між рішеннями:

– S3Proху для невеликих і середніх файлів (до 10 МБ), де важлива висока швидкість передавання;

– Apache Libcloud, коли важливим є зниження вартості;

– Apache NiFi, якщо необхідна більш складна логіка передачі.

Для оптимізації використання мультимарних підходів при наявності параметрів швидкодії і вартості пропонується наступний алгоритм:

1. Визначення вхідних параметрів. У нашому випадку це: розмір файлу, об'єм зберігання, критерій вартості, критерій часу доступу до файлу.

Для  $CC$  і  $CT$ , який параметр є більш важливим, призначаємо 1, інакше – 0.

2. Порівнюємо  $CC$  і  $CT$ .

3. Обираємо оптимальний підхід при  $CT=1$ , використовуючи функцію, що визначає середній час доступу для підходу  $A$  і заданого розміру файлу  $FS$ :

$$A_{opt} = \arg \min_A T(A, FS). \quad (5)$$

4. Обираємо оптимальний підхід при  $CC=1$ , використовуючи функцію, що визначає вартість для підходу  $A$  і заданого об'єму зберігання  $V$ :

$$A_{opt} = \arg \min_A C(A, V). \quad (6)$$

Для підходів Apache Libcloud і S3Proху правило прийняття рішень буде мати наступний вигляд:

ЯКЩО  $FS \leq 1$  МБ ТА  $CT=1$ ,

ТО обираємо S3Proху,

ІНАКШЕ обираємо Apache Libcloud.

Для опрацювання більш складних сценаріїв можна обрати Apache NiFi.

Алгоритм в загальному вигляді представлено на рис. 7.

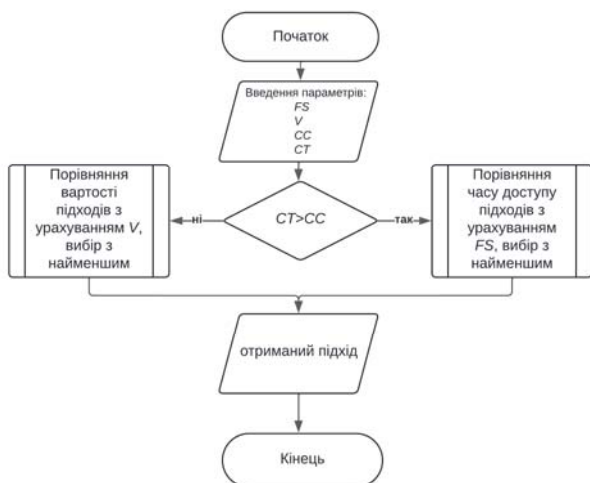


Рисунок 7 – Загальний алгоритм вибору мультимарного підходу

## 6 ОБГОВОРЕННЯ

Проведені дослідження показали, що підхід хмарних універсальних бібліотек (Apache Libcloud) і мультимарних шлюзів (S3Proху), маючи незначний розрив у експериментальних даних, можуть бути рекомендовані до впровадження. Для вибору підходу під конкретну задачу варто звернути на особливості кожного з них.

Хмарні універсальні бібліотеки:

– Дозволяють розробникам писати код, який може взаємодіяти з різними хмарними сервісами без залежності від конкретного провайдера.

– Спрощують процес інтеграції і роботи з хмарними сервісами.

– Можуть не підтримувати всі функції, які доступні у конкретних хмарних сервісах.

Мультимарні шлюзи:

– Дозволяють інтегрувати різні хмарні сервіси в єдине сховище, що полегшує управління даними в різних середовищах.

– Використовуються для зниження складнощів міграції даних між різними хмарними платформами.

– Додаткові шари абстракції, які вводить шлюз, можуть призвести до зниження продуктивності.

– Можуть вимагати додаткових витрат на обслуговування та підтримку.

Але для всебічного аналізу означених підходів може бути недостатнім використання лише цих критеріїв для оцінки ефективності впровадження. Тому подальше дослідження пропонується виконувати у напрямку оцінки складності впровадження кожного з рішень і побудови багатокритеріальної моделі визначення ефективності.

## ВИСНОВКИ

В ході роботи було досліджено такі підходи мультимарного доступу до даних:

– multi-cloud шлюзи;

– платформи керування даними;

– універсальні хмарні бібліотеки.

Для кожного з підходів було обрано технологію, на основі якої і проводився експеримент.

Наукова новизна. Встановлено, що універсальні хмарні бібліотеки, представлені Apache Libcloud, демонструють найкращі показники з точки зору вартості та продуктивності. Трохи поступається експериментальними даними S3Proху, який є представником мультимарних шлюзів. Найгірші показники має Apache NiFi (платформи керування даними).

Практична значимість дослідження. Результати дослідження можуть бути використані організаціями для вибору найбільш відповідного підходу до мультимарного доступу до даних відповідно до їх конкретних потреб і умов.

Подальші дослідження пропонується спрямувати на визначення критерію складності впровадження кожного з підходів. І далі, на основі усіх зазначених критеріїв, пропонується створити модель, яка б змогла окреслити методологію вибору найбільш ефективних інструментів.

## ЛІТЕРАТУРА

1. Petcu D. Multi-Cloud: Expectations and Current Approaches / D. Petcu // MultiCloud 2013 – Proceedings of the International Workshop on Multi-Cloud Applications and Federated Clouds – 2013. – P. 1–6. DOI: 10.1145/2462326.2462328.
2. Saceres A. State-of-the-Art Architectures for Interoperability of Heterogeneous Clouds / A. Saceres, L. S. Globa // Telecommunications and Computer Engineering (TCSET): 2022 IEEE 16th International



- Conference on Advanced Trends in Radioelectronics, Lviv-Slavske, Ukraine, 22–26 February 2022, : proceedings. – Los Alamitos: IEEE, 2022. – P. 704–709. DOI: 10.1109/TCSET55632.2022.9766965.
3. Ferrera A. J. Multi-Cloud Platform-as-a-Service Model, Functionalities and Approaches / A. J. Ferrera, D. G. Perez, R. S. Gonzalez // *Procedia Computer Science*. – 2016. – Vol. 97. – P. 63–72.
  4. Goasduff L. Why Organizations Choose a Multicloud Strategy [Electronic resource] / L. Goasduff. – Access mode: <https://www.gartner.com/smarterwithgartner/why-organizations-choose-a-multicloud-strategy>.
  5. Flexera 2024 State of the Cloud Report [Electronic resource]. – Access mode: <https://info.flexera.com/CM-REPORT-State-of-the-Cloud>.
  6. Козіна О. А. Архітектура проміжного програмного забезпечення для узгодження даних в мультихмарних системах / О. А. Козіна, В. І. Панченко, О. М. Рисованій // Вісник Національного технічного університету «ХПІ». – 2021. – № 2 (6). – С. 101–112. DOI: 10.20998/2411-0558.2021.02.07.
  7. Multi-Cloud: A Comprehensive Review / [H. A. Imran, U. Latif, A. A. Ikram et al.] // 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 05–07 November 2020: proceedings. – Los Alamitos: IEEE, 2021. – P. 1–5. DOI: 10.1109/INMIC50486.2020.9318176.
  8. Interoperability in the Heterogeneous Cloud Environment: A Survey of Recent User-Centric Approaches / [I. Mansour, R. Sahandi et al.] // ICC '16: Proceedings of the International Conference on Internet of things and Cloud Computing, Cambridge, United Kingdom, 22–23 March 2016: proceedings. – New York, NY, USA: Association for Computing Machinery. – P. 1–7. DOI: 10.1145/2896387.2896447.
  9. Framework Semantic and Standard Approaches in Multi-clouds to Achieve Interoperability: A Survey / [Z. A. Adhoni, N. Dayanand Lal et al.] // *Journal of Integrated Science and Technology*. – 2022. – Vol. 10, No. 2. – P. 67–72.
  10. Sheldon R. Cloud Data Management Interface (CDMI) [Electronic resource] / R. Sheldon. – Access mode: <https://www.techtarget.com/searchstorage/definition/Cloud-Data-Management-Interface>.
  11. What is Open Cloud Computing Interface (OCCI) [Electronic resource]. – Access mode: <https://www.analyticssteps.com/blogs/what-open-cloud-computing-interface-occi>.
  12. Bennett S. Cloud Storage Gateway Vendors: 7 Top Solutions [Electronic resource] / S. Bennett. – Access mode: <https://www.resilio.com/blog/cloud-storage-gateway-vendors>.
  13. Data Management Platforms: An Empirical Taxonomy / [J. A. Hüllmann, A. Sivakumar, S. Krebber et al.] // 34th Bled eConference, Bled, Slovenia, 27–30 June 2021: proceedings. Available at: <https://aisel.aisnet.org/bled2021/9>.
  14. Integrated Management of IaaS Resources / [F. Meireles, B. Malheiro et al.] // European Conference on Parallel Processing. Euro-Par 2014: Parallel Processing Workshops. – 2014. – Cham : Springer, 2014. – P. 73–84. DOI: 10.1007/978-3-319-14313-2\_7.
  15. Malinina J. Cloud Agnostic – an Approach You Might Want to Avoid [Electronic resource] / J. Malinina. – Access mode: <https://medium.com/coderbyte/cloud-agnosticism-and-its-hidden-cost-4d3ed6d963f>.
  16. Parker E. S3 File Gateway for Efficient, Multi-Cloud File Access [Electronic resource] / E. Parker. – Access mode: <https://www.resilio.com/blog/s3-file-gateway>.
  17. Yousry A. Data Management: A Guide to Apache NiFi [Electronic resource] / A. Yousry. – Access mode: <https://medium.com/@ansam.yousry/data-management-a-guide-to-apache-nifi-21a29ecc4591>.
  18. Apache Libcloud [Electronic resource]. – Access mode: <https://libcloud.apache.org/>.
  19. Interoperability and Portability for Cloud Computing: A guide [Electronic resource]. – Access mode: <https://www.omg.org/cloud/deliverables/CSCC-Interoperability-and-Portability-for-Cloud-Computing-A-Guide.pdf>.
  20. AWS Pricing [Electronic resource]. – Access mode: [https://aws.amazon.com/ru/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=\\*all&awsf.tech-category=\\*all](https://aws.amazon.com/ru/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all).
  21. Docker Image for S3-Proxy. Reverse proxy for AWS S3 with basic authentication [Electronic resource]. – Access mode: <https://hub.docker.com/r/pottava/s3-proxy>.
  22. Caceres A. Link to GitHub repository with experiment s3Proxy [Electronic resource] / A.Caceres. – Access mode: [https://github.com/MA3STR0/multicloud\\_experiments/tree/master/multicloud\\_storage/s3proxy](https://github.com/MA3STR0/multicloud_experiments/tree/master/multicloud_storage/s3proxy).
  23. Docker Image for Apache NiFi. Apache NiFi unofficial binary build [Electronic resource]. – Access mode: <https://hub.docker.com/r/apache/nifi>.
  24. Caceres A. Link to GitHub repository with experiment Apache NiFi [Electronic resource] / A.Caceres. – Access mode: [https://github.com/MA3STR0/multicloud\\_experiments/tree/master/multicloud\\_storage/apache\\_nifi](https://github.com/MA3STR0/multicloud_experiments/tree/master/multicloud_storage/apache_nifi).
  25. Repository Apache Libcloud [Electronic resource]. – Access mode: <https://github.com/apache/libcloud>.
  26. Caceres A. Link to GitHub repository with experiment Libcloud [Electronic resource] / A.Caceres. – Access mode: [https://github.com/MA3STR0/multicloud\\_experiments/tree/master/multicloud\\_storage/libcloud](https://github.com/MA3STR0/multicloud_experiments/tree/master/multicloud_storage/libcloud).

Received 30.08.2024.  
Accepted 14.01.2025.

## ANALYSIS OF DATA ACCESS APPROACHES IN A MULTI-CLOUD ENVIRONMENT

**Caceres A.** – Postgraduate student, Department of Information Technologies in Telecommunications, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine.

**Globa L.** – Professor, Dr. Sc., Department of Information Technologies in Telecommunications, Institute of Telecommunication Systems, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine.

### ABSTRACT

**Context.** A multi-cloud system is characterized by the sequential or simultaneous use of services from different cloud providers to run applications. Such a system is a preferred infrastructure for the vast majority of IT businesses today. Currently, there are various approaches to combining cloud platforms from multiple vendors. This article explores practical approaches to achieve multi-cloud interoperability, focusing on abstract data access between different cloud storage providers and multi-cloud computing resource allocation. Key technologies and methodologies for uninterrupted data management are presented, such as the use of multi-cloud storage gateways (using S3Proxy as an example), the implementation of data management platforms (Apache NiFi), and the use of cloud-agnostic libraries (Apache Libcloud). The paper highlights the advantages and disadvantages of the selected approaches and conducts experiments to determine the cost and performance of these strategies. The result of the research is to determine the cost and performance of different approaches to data access in multi-cloud environments.

**Objective.** To investigate different approaches to multi-cloud data access and determine the most optimal in terms of cost and performance.

**Method.** We propose the optimization of multi-cloud infrastructures based on experimental data. Experimental modeling includes empirical measurements of performance and comparison of storage costs. The determination of performance is based on the measurement of data reading time and latency. The AWS S3 pricing model is used to estimate the cost. Optimization approaches are described, considering file sizes and data storage, combining the strengths of different multi-cloud approaches and dynamic switching between solutions. An algorithm for selecting multi-cloud approaches is proposed, which takes into account the criteria of cost and performance, as well as their priority.

**Results.** The experiment yielded values for the cost of storing and downloading data of different sizes (100 GB, 1 TB, 10 TB), and the performance of transferring files of different sizes (100 KB, 1 MB, 10 MB) for multi-cloud gateway technologies, data management platforms, and cloud-agnostic libraries. S3Proxy was found to have the fastest file access for large data volumes, while Apache Libcloud showed better value for smaller volumes. Both approaches significantly outperformed Apache NiFi. This study can contribute to the development of methods for efficient resource management in multi-cloud environments.

**Conclusions.** The obtained results can assist in prioritizing the selection of these paradigms, aiding organizations in developing and deploying effective multi-cloud strategies. This approach enables them to leverage the distinctive features of each cloud provider while maintaining a unified, flexible, and efficient storage and computing environment.

**KEYWORDS:** cloud computing, multi-cloud access, cloud-agnostic libraries.

### REFERENCES

1. Petcu D. Multi-Cloud: Expectations and Current Approaches, *MultiCloud 2013 – Proceedings of the International Workshop on Multi-Cloud Applications and Federated Clouds*, 2013, pp. 1–6. DOI: 10.1145/2462326.2462328.
2. Caceres A., Globa L. S. State-of-the-Art Architectures for Interoperability of Heterogeneous Clouds, *Telecommunications and Computer Engineering (TCSET): 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Lviv-Slavske, Ukraine, 22–26 February 2022, : proceedings*. Los Alamitos, IEEE, 2022, pp. 704–709. DOI: 10.1109/TCSET55632.2022.9766965.
3. Ferrera A. J., Perez D. G., Gonzalez R. S. Multi-Cloud Platform-as-a-Service Model, Functionalities and Approaches, *Procedia Computer Science*, 2016, Vol. 97, pp. 63–72.
4. Goasduff L. Why Organizations Choose a Multicloud Strategy [Electronic resource]. Access mode: <https://www.gartner.com/smarterwithgartner/why-organizations-choose-a-multicloud-strategy>.
5. Flexera 2024 State of the Cloud Report [Electronic resource]. Access mode: <https://info.flexera.com/CM-REPORT-State-of-the-Cloud>.
6. Kozina O. A., Panchenko V. I., Risovaniy O. M. Arkhitektura promizhnoho prohramnoho zabezpechennya dlya uzgodzhennya danykh v mul'tykhmarinykh systemakh, *Visnik Nacionalnogo tehnicnogo universitetu “HPI”*, 2021, No. 2 (6), pp. 101–112. DOI: 10.20998/2411-0558.2021.02.07.
7. Imran H. A., Latif U., Ikram A. A. et al. Multi-Cloud: A Comprehensive Review, *2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 05–07 November 2020: proceedings*. Los Alamitos, IEEE, 2021, pp. 1–5. DOI: 10.1109/INMIC50486.2020.9318176.
8. Mansour I., Sahandi R. et al. Interoperability in the Heterogeneous Cloud Environment: A Survey of Recent User-Centric Approaches, *ICC '16: Proceedings of the International Conference on Internet of things and Cloud Computing, Cambridge, United Kingdom, 22–23 March 2016: proceedings*. New York, NY, USA: Association for Computing Machinery, pp. 1–7. DOI: 10.1145/2896387.2896447.

9. Adhoni Z. A., Dayanand Lal N. et al. Framework Semantic and Standard Approaches in Multi-clouds to Achieve Interoperability: A Survey, *Journal of Integrated Science and Technology*, 2022, Vol. 10, No. 2, pp. 67–72.
10. Sheldon R. Cloud Data Management Interface (CDMI) [Electronic resource]. Access mode: <https://www.techtarget.com/searchstorage/definition/Cloud-Data-Management-Interface>.
11. What is Open Cloud Computing Interface (OCCI) [Electronic resource]. Access mode: <https://www.analyticssteps.com/blogs/what-open-cloud-computing-interface-occi>.
12. Bennett S. Cloud Storage Gateway Vendors: 7 Top Solutions [Electronic resource]. Access mode: <https://www.resilio.com/blog/cloud-storage-gateway-vendors>.
13. Hüllmann J. A., Sivakumar A., Krebber S. et al. Data Management Platforms: An Empirical Taxonomy, *34th Bled eConference, Bled, Slovenia, 27–30 June 2021: proceedings*. Available at: <https://aisel.aisnet.org/bled2021/9>.
14. Meireles F., Malheiro B. et al. Integrated Management of IaaS Resources, *European Conference on Parallel Processing. Euro-Par 2014: Parallel Processing Workshops, 2014*. Cham, Springer, 2014, pp. 73–84. DOI: 10.1007/978-3-319-14313-2\_7.
15. Malinina J. Cloud Agnostic – an Approach You Might Want to Avoid [Electronic resource]. Access mode: <https://medium.com/coderbyte/cloud-agnosticism-and-its-hidden-cost-4d3ed6d963f>.
16. Parker E. S3 File Gateway for Efficient, Multi-Cloud File Access [Electronic resource]. Access mode: <https://www.resilio.com/blog/s3-file-gateway>.
17. Yousry A. Data Management: A Guide to Apache NiFi [Electronic resource]. Access mode: <https://medium.com/@ansam.yousry/data-management-a-guide-to-apache-nifi-21a29ecc4591>.
18. Apache Libcloud [Electronic resource]. Access mode: <https://libcloud.apache.org/>.
19. Interoperability and Portability for Cloud Computing: A guide [Electronic resource]. Access mode: <https://www.omg.org/cloud/deliverables/CSCC-Interoperability-and-Portability-for-Cloud-Computing-A-Guide.pdf>.
20. AWS Pricing [Electronic resource]. Access mode: [https://aws.amazon.com/ru/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=\\*all&awsf.tech-category=\\*all](https://aws.amazon.com/ru/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all).
21. Docker Image for S3-Proxy. Reverse proxy for AWS S3 with basic authentication [Electronic resource]. Access mode: <https://hub.docker.com/r/pottava/s3-proxy>.
22. Caceres A. Link to GitHub repository with experiment s3Proxy [Electronic resource]. Access mode: [https://github.com/MA3STR0/multicloud\\_experiments/tree/master/multicloud\\_storage/s3proxy](https://github.com/MA3STR0/multicloud_experiments/tree/master/multicloud_storage/s3proxy).
23. Docker Image for Apache NiFi. Apache NiFi unofficial binary build [Electronic resource]. Access mode: <https://hub.docker.com/r/apache/nifi>.
24. Caceres A. Link to GitHub repository with experiment Apache NiFi [Electronic resource]. Access mode: [https://github.com/MA3STR0/multicloud\\_experiments/tree/master/multicloud\\_storage/apache\\_nifi](https://github.com/MA3STR0/multicloud_experiments/tree/master/multicloud_storage/apache_nifi).
25. Repository Apache Libcloud [Electronic resource]. Access mode: <https://github.com/apache/libcloud>.
26. Caceres A. Link to GitHub repository with experiment Libcloud [Electronic resource]. Access mode: [https://github.com/MA3STR0/multicloud\\_experiments/tree/master/multicloud\\_storage/libcloud](https://github.com/MA3STR0/multicloud_experiments/tree/master/multicloud_storage/libcloud).