

Д. А. Пасько, Г. І. Молчанов, В. В. Давидов

Національний технічний університет «Харківський політехнічний інститут», Харків, Україна

УПРАВЛІННЯ НЕОБМЕЖЕНОЮ КІЛЬКІСТЮ ХМАРНИХ СХОВИЩ

Предметом статті є дослідження можливості створення зручного механізму управління процесом збереження даних з використанням необмеженої кількості хмарних сховищ. **Метою** є дослідження методів оптимізації і спрощення управління процесом збереження даних з використанням необмеженої кількості хмарних сховищ. **Завдання:** розробити архітектуру програмного комплексу для управління розміщенням даних у найпоширеніших хмарних сховищах; обрати ефективний алгоритм роботи при виборі сховища для збереження даних; виконати тестування програмного продукту. Використовуваними **методами** є: спостереження за пропускну здатністю хмарних файлових сховищ; порівняння зручності використання функціоналу хмарних сервісів; вимір завантаженості сховища та перевірка наявності вільного простору; експеримент по створенню програми управління хмарними сховищами. Отримано такі **результати**. Обґрунтовано спосіб вибору сховища для вивантаження даних. Розроблено програмне забезпечення, яке надає можливість об'єднати необмежену кількість хмарних сховищ у єдиний простір та надати можливість зручного управління та надання необхідної інформації для користувача. Результатом дослідження є розробка програмного комплексу, який виступає в ролі концентратора для хмарних сховищ, а також реалізація методів вибору сховища для вивантаження файлів, заснованих на оцінці проценту вільного простору кожного сховища та функції балансування завантаженості сховищ даними. **Висновки.** Наукова новизна отриманих результатів полягає в наступному: досліджено існуючі методи збереження даних у хмарних сховищах; удосконалено метод гібридного збереження даних за рахунок об'єднання необмеженої кількості сховищ у єдиний простір під централізованою системою управління. Розроблено систему вибору сховища для вивантаження файлів, що заснована на оцінці та порівнянні проценту вільного простору, частоти використання і середнього часу доступу. Розроблено алгоритм балансування завантаженості хмарних сховищ даними. Проведені експериментальні дослідження підтверджують можливість використання запропонованого підходу для зручного управління необмеженою кількістю хмарних сховищ.

Ключові слова: хмарні сховища; гібридне хмарне сховище; хмарні технології; Google Drive; Google Drive API; Mega; Mega API; публічні хмари; об'єднання хмарних сховищ.

Вступ

Постановка проблеми. У наш час розвиток цифрових технологій не стоїть на місці. І щоб відповідати сучасним стандартам потрібно постійно вдосконалювати обладнання і методи зберігання інформації. Також постійно виникає питання зручності та доступності використання даних. Дані можуть зберігатися в локальних сховищах, на жорстких дисках комп'ютерів, на переносних пристроях або в хмарних сховищах. Зберігання даних в «хмарах» вирішує питання зручності та доступності, тому що в такому випадку дані стають доступні з будь-якої точки світу, де є підключення до Інтернету. В разі використання хмарних сервісів всі дані зберігаються на віддалених серверах, а користувач, маючи гарну швидкість Інтернету, може працювати з ними так, ніби вони перебувають у нього на комп'ютері.

Способи організації хмарного сховища бувають різні, але суть у всіх одна – надати користувачу віддалений доступ до даних. Хмарні сервіси бувають публічними, приватними або гібридними. Приватні сервіси дозволяють користувачу налаштувати хмарне сховище повністю під себе, легко регулюючи обсяги і будь-які інші параметри, такі як пропускну здатність і тому подібне. Публічні сервіси накладають обмеження на управління хмарою і завжди мають обмежений безкоштовний обсяг пам'яті. Гібридні ж, в свою чергу, об'єднують в одну систему і приватні і публічні хмари з метою балансування навантаження на приватне сховище, за рахунок використання публічного, або для резервного копіювання деяких даних.

Аналіз літератури [1, 2] показав, що існує багато онлайн-сервісів, які надають безкоштовно деякий обсяг простору. Зазвичай це близько 10 гігабайт, і для використання більшого обсягу простору треба платити щомісяця, в середньому, близько 10 доларів. До переваг існуючих методів збереження інформації у хмарах можна віднести надійність, своєчасність отримання і доступність даних в «хмарі», яка дуже сильно залежить від багатьох проміжних параметрів, таких як: канали передачі даних на шляху від клієнта до «хмари», надійність останньої милі, якість роботи Інтернет-провайдерів клієнта, доступність самої «хмари» в даний момент часу. До недоліків можна віднести щомісячну абонентську плату за використання сервісів, а також загальну продуктивність при роботі з даними в «хмарі», яка може бути нижче, ніж при роботі з локальними копіями даних.

Основним методом усунення зазначених недоліків є вдосконалення технології хмарного збереження даних за рахунок балансування заповнення хмарних сховищ інформацією. Також, для покращення швидкості обробки файлів, вводиться кешування файлів, доступ до яких найчастіший. Для вирішення проблем з абонентською платою можна використовувати цей програмний комплекс.

У [3, 4] наведено лише опис вже існуючих технологій та способів збереження даних у хмарах, що лягли в основу комплексу об'єднання сховищ.

Метою статті є дослідження можливості оптимізації і спрощення управління процесом збереження даних з використанням необмеженої кількості хмарних сховищ.

Результати розробки та досліджень

Для розробки програмного комплексу управління процесом збереження даних було використано два хмарних сервіси: Google Drive та Mega.

Google Drive – це служба зберігання і синхронізації файлів, розроблена компанією Google [5]. Google Диск дозволяє користувачам зберігати файли на своїх серверах, синхронізувати файли на різних пристроях і обмінюватися файлами. Google Диск пропонує користувачам 15 гігабайт безкоштовного сховища та інші розміри, пропоновані за допомогою додаткових платних планів. Щоб підключити користувача до системи, потрібно включити використання API для його облікового запису [6], створивши ідентифікатор клієнта OAuth 2. Після цього, обліковий запис можна додавати до програми разом з раніше створеним файлом ідентифікатора.

Mega – файлообмінник з шифруванням даних [7]. Mega шифрує весь контент прямо в браузері за допомогою алгоритму AES. Користувачі можуть передавати один одному файли в зашифрованому вигляді, при цьому всі дані зберігаються в «хмарі». Mega надає 15 ГБ безкоштовного розміру (50 ГБ у перший місяць користування), а також інші платні плани для збільшення розміру сховища. Підключення до програми виконується введенням логіну і паролю від облікового запису до програми.

Для об'єднання сховищ у єдину систему, спочатку потрібно було вирішити питання зберігання структури даних, у якому вигляді представляти та відображувати файли. Зберігати файли було вирішено у тому виді, в якому вони завантажувалися з хмарних сервісів – у вигляді списків. Відображувати файли було вирішено в одному з найпоширеніших способів – у вигляді дерева. Кожен файл має інформацію про «родичів», тому легко можна створити дерево. Дерево створюється за допомогою рекурсії, виконується пошук для кожної папки кореневого каталогу його дітей, та створення нової гілки, після цього рекурсивно виконується пошук дітей для кожного попереднього елемента і так далі, поки дерево не буде створено.

При об'єднанні сховищ, ми будемо використовувати раніше створені дерева для кожного сховища. Об'єднання сховищ робиться теж рекурсивно, як і створення дерев. При об'єднанні не створюється нового дерева, а додаються нові гілки у дерево головного сховища (яке було обрано заздалегідь). Завдяки цьому не потрібно рекурсивно продивлятися усі гілки сховища, з яким робиться об'єднання, тому що, при додаванні у головне дерево, додається гілка з усією своєю структурою і дітьми.

Алгоритм об'єднання:

1. Перевіряємо, чи є у кореневому каталозі головного сховища папка з такою ж назвою, як і в сховищі донорі.

2. Якщо така папка існує, то робимо помітку в структурі цієї папки головного сховища, указуючи, що вона належить обом сховищам (додаємо логін сховища, якому вона теж належить) і переходимо до кроку №3. Якщо ж такої папки не існує, просто додаємо її до головного сховища і переходимо порівнюва-

ти іншу папку сховища-донора з папкою сховища-реципієнта.

3. Рекурсивно передаємо дітей обох сховищ (головного і донора) до цього методу, тобто повертаємося на крок №2, але вже порівнюємо дітей, тобто спускаємося на рівень вниз по дереву. Коли ми дійдемо до кінця дерева, тобто дітей не залишиться, алгоритм закінчується.

Об'єднання дерев робиться, доки не буде зроблено об'єднання з усіма деревами. Після цього, на екран користувача виводиться єдина структура дерева з папками з усіх сховищ.

Алгоритм загрузки файлів у сховища складається з двох етапів. Перший - користувач обирає необхідні файли для відправлення у сховища, та перетягує їх у зони відправлення файлів (рис. 1), у якій вони відображуються після додавання.

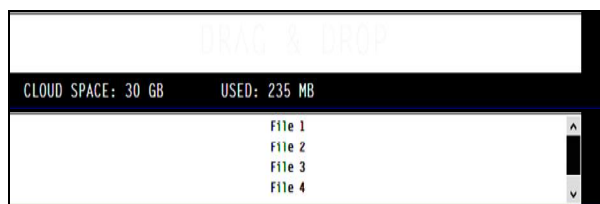


Рис. 1. Файли у зоні відправлення

Після цього, настає другий етап – оцінка сховища, у яке буде відправлено файли. Етап ґрунтується на оцінці проценту вільного простору на диску. Оцінюється не розмір вільної пам'яті, а процент вільного простору, тому що у хмарних сервісів може бути різний розмір пам'яті, яку вони надають користувачу. Таким чином легко можна балансувати завантаження даними усіх сховищ.

На рис. 2 можна побачити завантаженість даними у процентному відношенні сховищ, які використовуються у програмі, а у табл. 1 наведено інформацію про вільне місце у гігабайтах.

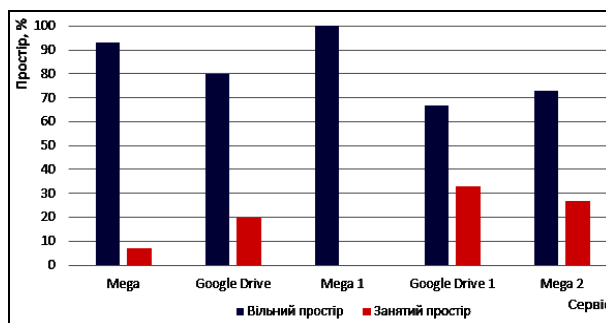


Рис. 2. Стан сховищ до завантаження файлів

Таблиця 1. Стан простору сховищ до завантаження файлів

Сховище	Вільний простір, ГБ	Занятий простір, ГБ
Mega	14	1
Google Drive	12	3
Mega 1	15	0
Google Drive 1	10	5
Mega 2	11	4

Після того, як було отримано інформацію про завантаженість сховищ, файли починають відправлятися до сховищ (рис. 3).

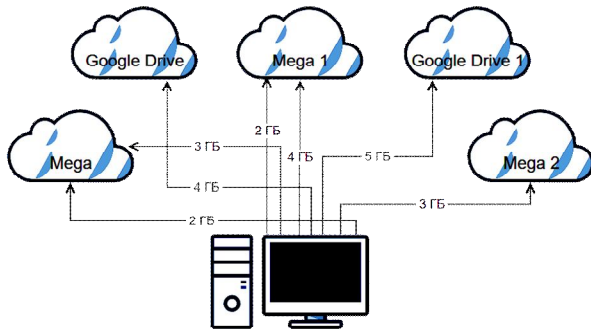


Рис. 3. Почергове відправлення файлів у сховища

Після відправлення 7 файлів розміром 2 ГБ, 3 ГБ, 4 ГБ, 4 ГБ, 5 ГБ, 3 ГБ, 2 ГБ, наповненість сховищ виглядає наступним чином (див. рис. 4), а у табл. 2 можна побачити стан простору сховищ після завантаження файлів.

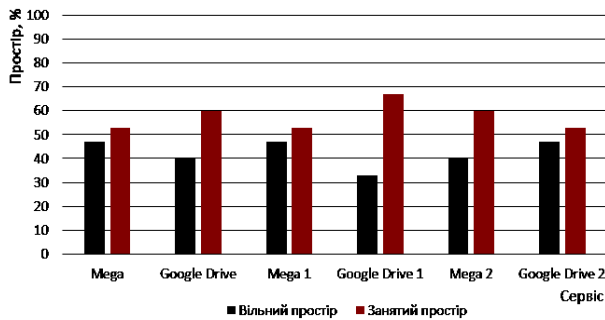


Рис. 4. Стан сховищ після завантаження файлів

Таблиця 2. Стан простору сховищ після завантаження файлів

Сховище	Вільний простір, ГБ	Занятий простір, ГБ
Mega	9	6
Google Drive	8	7
Mega 1	9	6
Google Drive 1	5	10
Mega 2	8	7

Можна побачити, що алгоритм розподілення даних по сховищам, майже рівномірно заповнив усі сховища файлами. На етапі відправлення 5-го та 6-го файлів декілька сховищ мали рівний вільний обсяг, але файли були відправлені до сховища Google Drive 1 та Mega 2 відповідно, тому що до цього моменту туди нічого не відправлялося.

Алгоритм оцінки вільного простору при відправленні файлів було обрано, тому що для балансування завантаженості всіх сховищ, які підключені до програми, цей спосіб є оптимальним. Він надає можливість рівномірно розподіляти дані, та допомагає зменшити вплив однієї з проблем використання хмарних сховищ, а саме – проблему недоступності схо-

вища. Коли дані розподілені рівномірно, то зменшується шанс втратити усю інформацію, як в ситуації, якщо вона би зберігалася лиш на одному сховищі. При недоступності хмарного диску (технічні проблеми, загублення паролю) ми будемо мати доступ до інших сховищ, і до інформації, яка там розподілена.

Також, було розроблено алгоритм балансування розподілення даних в разі, коли додається новий акаунт до системи. Алгоритм надає можливість збалансувати навантаження на усі сховища, якщо один з облікових записів перевантажений чи вільний.

Додамо до нашої системи повністю заповнений акаунт Google Drive сховища (рис. 5), в табл. 3 наведено стан розміру сховищ після додавання облікового запису.

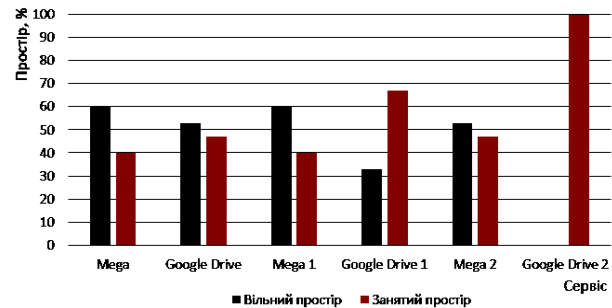


Рис. 5. Заповнення сховищ після додавання нового облікового запису

Таблиця 3. Стан простору сховищ після додавання нового облікового запису

Сховище	Вільний простір, ГБ	Занятий простір, ГБ
Mega	9	6
Google Drive	8	7
Mega 1	9	6
Google Drive 1	5	10
Mega 2	8	7
Google Drive 2	0	15

Для балансування навантаження даними у системі, кількість даних, які потрібно відправити в інші сховища, з округленням в менший бік буде розраховано за формулою: $S = M / N$, де M – розмір даних, які потрібно розподілити по сховищам, а N – кількість хмарних сховищ.

Після того, як розмір даних, які потрібно відправити у кожне сховище розраховано, починається відправка файлів до інших сховищ. Вона відбувається, доки завантаженість хмарних дисків не буде майже рівною. За формулою, програма буде розподіляти по сховищам по 2 ГБ даних. Схематичний процес балансування зображено на рис. 6.

Результат балансування зображено на рис. 7, а у табл. 4 можна побачити стан розміру сховищ після балансування.

В результаті балансування, дані були відправлені в Mega 1 – 2 ГБ, Mega – 2 ГБ, Google Drive – 2 ГБ, та Mega 2 – 2 ГБ.

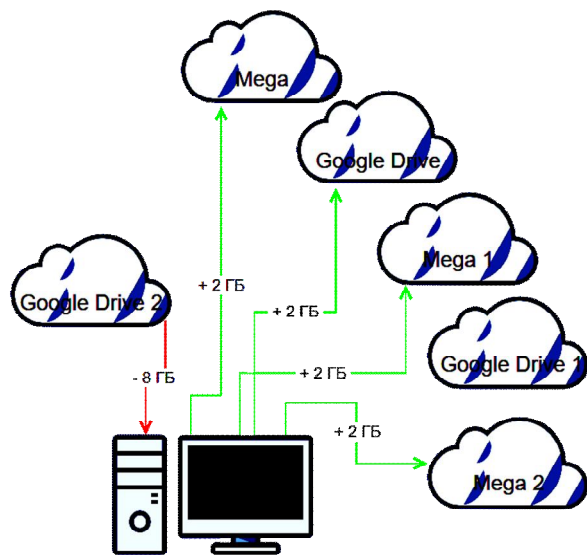


Рис. 6. Балансування завантаженості хмарних сховищ

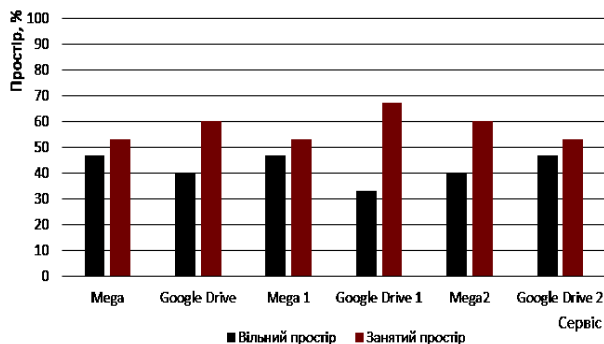


Рис. 7. Стан сховищ після балансування файлів

Таблиця 4. Стан простору сховищ після балансування даних

Сховище	Вільний простір, ГБ	Занятий простір, ГБ
Mega	7	8
Google Drive	6	9
Mega 1	7	8
Google Drive 1	5	10
Mega 2	6	9
Google Drive 2	7	8

Висновки

В роботі виконано дослідження можливості створення централізованої системи управління хмарними сховищами та розроблено програмний комплекс для об'єднання необмеженої кількості найпоширеніших хмарних сервісів Google Drive та Mega. На рис. 8 зображено інтерфейс програми.

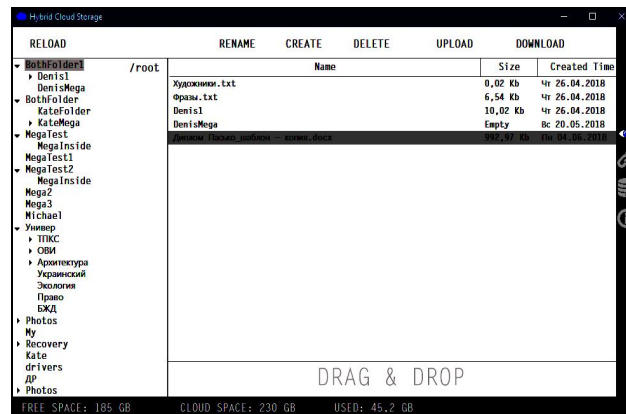


Рис. 8. Інтерфейс програмного комплексу для управління сховищами

Обґрунтовано вибір алгоритму управління процесом збереження даних.

Удосконалено існуючі алгоритми збереження даних у хмарних сховищах за рахунок можливості об'єднання необмеженої кількості облікових записів сховищ та балансування завантаженості кожного за рахунок реалізації алгоритму оцінювання процентного відношення місця у сховищах.

Був розроблений та реалізований алгоритм поєднання сховищ для зручного управління та відображення файлів користувача.

Розроблено алгоритм балансування завантаженістю даними сховищ при додаванні нового облікового запису, як незаповненого так і вже заповненого даними. Балансування можна виконати за бажанням користувача.

Проведені дослідження підтверджують, що наведена організація сховищ підвищує стабільність збереження даних, а також надає можливість мати великий розмір хмарного сховища безкоштовно.

СПИСОК ЛІТЕРАТУРИ

1. Tucha [Електронний ресурс]: Гібридне сховище. – Режим доступу: <http://tucha.ua/blog/gibridnoe-oblako/>
2. Сервіоніка [Електронний ресурс]: Гібридне сховище. – Режим доступу: <http://servionica.ru/services/138/>
3. ZDNet [Електронний ресурс]: Hybrid cloud: What it is, why it matters. – Режим доступу: <http://www.zdnet.com/article/hybrid-cloud-what-it-is-why-it-matters/>
4. The NIST Definition of Cloud Computing – Mell, Peter and Grance, Timothy, 2011 – 6 с.
5. Wikipedia [Електронний ресурс]: – Google Drive. – Режим доступу: https://en.wikipedia.org/wiki/Google_Drive
6. Google [Електронний ресурс]: – Enable API. – Google. Дата звернення: 17.05.2018. – Режим доступу: <https://console.developers.google.com/flows/enableapi?apiid=drive>
7. Wikipedia [Електронний ресурс]: – Mega. Дата звернення: 13.05.2018. – Режим доступу: [https://en.wikipedia.org/wiki/Mega_\(service\)](https://en.wikipedia.org/wiki/Mega_(service))
8. Google [Електронний ресурс]: – Google Drive API v3. – Google. Дата звернення: 14.05.2018. – Режим доступу: <https://developers.google.com/drive/api/v3/quickstart/dotnet>
9. Mega [Електронний ресурс]: – Mega cloud storage API. – MEGA. Дата звернення: 14.05.2018. – Режим доступу: <https://gpailler.github.io/MegaApiClient/articles/faq.html>
10. Інтеграція – основа хмари. Відкриті системи. СУБД – Леонід Черняк, 2011.

REFERENCES

1. Tucha, *Hybrid storage*, available at: <http://tucha.ua/blog/gibridnoe-oblako/> (last accessed March 15, 2018).
2. Servio, *Hybrid storage*, available at: <http://servionica.ru/services/138/> (last accessed March 15, 2018).
3. ZDNet, *Hybrid cloud: What is it, why it matters*, available at: <http://www.zdnet.com/article/hybrid-cloud-what-it-is-why-it-matters/> (last accessed March 15, 2018).
4. Mell, Peter and Grance, Timothy (2011), *The NIST Definition of Cloud Computing*, 6 p.
5. Wikipedia, *Google Drive*, available at: https://en.wikipedia.org/wiki/Google_Drive (last accessed March 15, 2018).
6. Google, *Enable API Google*, available at: <https://console.developers.google.com/flows/enableapi?apiid=drive> (last accessed March 15, 2018).
7. Wikipedia, *Mega*, available at: [https://en.wikipedia.org/wiki/Mega_\(service\)](https://en.wikipedia.org/wiki/Mega_(service)) (last accessed March 15, 2018).
8. Google, *Google Drive API v3*, available at: <https://developers.google.com/drive/api/v3/quickstart/dotnet> (last accessed March 15, 2018).
9. Mega, *Mega cloud storage API. – MEGA*, available at: <https://gpallier.github.io/MegaApiClient/articles/faq.html> (last accessed March 15, 2018).
10. Leonid Chernyak (2011), *Integration is the basis of the cloud. Open systems. DBMS*.

Надійшла (received) 13.07.2018

Прийнята до друку (accepted for publication) 29.08.2018

Управление неограниченным количеством облачных хранилищ

Д. А. Пасько, Г. И. Молчанов, В. В. Давыдов

Предметом статьи является исследование возможности создания удобного механизма управления процессом хранения данных с использованием неограниченного количества облачных хранилищ. **Целью** является исследование методов оптимизации и упрощения управления процессом хранения данных с использованием неограниченного количества облачных хранилищ. **Задача:** разработать архитектуру программного комплекса для управления размещением данных в самых распространенных облачных хранилищах; выбрать эффективный алгоритм работы при выборе хранилища для размещения данных; выполнить тестирование программного продукта. Используемыми **методами** являются: наблюдение за пропускной способностью облачных файловых хранилищ; сравнения удобства использования функционала облачных сервисов; измерение загруженности хранилища и проверка наличия свободного пространства; эксперимент по созданию программы управления облачными хранилищами. Получены следующие **результаты**. Обоснован способ выбора хранилища для выгрузки данных. Разработано программное обеспечение, которое позволяет объединить неограниченное количество облачных хранилищ в единое пространство, а также предоставляет возможность удобного управления и получения необходимой информации для пользователя. Результатом исследования является разработка программного комплекса, который выступает в роли концентратора для облачных хранилищ, а также реализация методов выбора хранилища для выгрузки файлов, основанных на оценке процента свободного пространства каждого хранилища и функции балансировки загруженности хранилищ данными. **Выводы.** Научная новизна полученных результатов заключается в следующем: исследованы существующие методы сохранения данных в облачных хранилищах; усовершенствован метод гибридного хранения данных за счет объединения неограниченного количества хранилищ в единое пространство под централизованной системой управления. Разработана система выбора хранилища для выгрузки файлов, основанная на оценке и сравнении процента свободного пространства, частоты использования и среднего времени доступа. Разработан алгоритм балансировки загруженности облачных хранилищ данными. Проведенные экспериментальные исследования подтверждают возможность использования предложенного подхода для удобного управления неограниченным количеством облачных хранилищ.

Ключевые слова: облачные хранилища; гибридное облачное хранилище; облачные технологии; Google Drive; Google Drive API; Mega; Mega API; публичные облака; объединение облачных хранилищ.

Unlimited cloud storage management

D. Pasko, H. Molchanov, V. Davydov

The **subject** of the article is the management of the data storage process using an unlimited number of cloud storages. The **goal** is to study the possibility of optimizing and simplifying the management of the data storage process using an unlimited number of cloud storages. **Objective:** develop the architecture of a software package for managing the placement of data in the most common cloud storages; select the most efficient storage selection algorithm for storing data; perform software testing. The **methods** used are: monitoring the throughput of cloud file storage; comparing the usability of the cloud services functionality; measuring storage load and checking for free space; experimental creation of cloud storage management software. The following **results** have been obtained. The storage selection mechanism for loading data is described. Developed software for uniting in a single space and convenient management of an unlimited number of cloud storage. The result of the research is the development of a software package that acts as a hub for cloud storage. Methods for selecting storage for uploading files based on an estimate of the percentage of free space of each storage and the function of load balancing of storage data are implemented. **Conclusions.** The scientific novelty of the results is as follows. Investigated the existing methods of storing data in the cloud storage. The method of hybrid data storage has been improved by combining an unlimited number of storages into a single space under a centralized management system. A system for selecting storage for uploading files has been developed which is based on estimating and comparing the percentage of free space with the frequency of use and average access time. An algorithm for balancing the workload of cloud storage has been developed. The conducted experimental studies confirm the possibility of using the proposed approach for convenient management of an unlimited number of cloud storages.

Keywords: cloud storage; hybrid cloud storage; cloud technology; Google Drive; Google Drive API; Mega; Mega API; public clouds; cloud storages union.