Ростислав П. Струбицький *(Національний університет «Львівська політехніка», Україна)*
Наталія Б. Шаховська *(Національний університет «Львівська політехніка», Україна)*

АНАЛІЗ ПІДХОДІВ ДО МОДЕЛЮВАННЯ ХМАРКОВИХ СХОВИЩ ДАНИХ*

У статті проведено аналіз підходів до розгортання сучасних хмаркових сховищ даних. Визначено основні напрямки реалізації розподілених хмаркових сховищ даних, основні труднощі їх розгортання. Запропоновано підходи до підвищення якості надання послуг такими сховищами.
Ключові слова: *сховища даних, хмаркові обчислення, розподілені сховища.*
Рис. 2. Літ. 11.

Ростислав П. Струбицкий *(Национальный университет «Львовская политехника», Украина)*
Наталия Б. Шаховская *(Национальный университет «Львовська политехника», Украина)*

АНАЛИЗ ПОДХОДОВ К МОДЕЛИРОВАНИЮ ОБЛАЧНЫХ ХРАНИЛИЩ ДАННЫХ

В статье проведен анализ подходов к развертыванию современных облачных хранилищ данных. Определены основные направления реализации распределенных облачных хранилищ данных, основные трудности их развертывания. Предложены подходы к повышению качества предоставления услуг такими хранилищами.
Ключовые слова: *хранилища данных, облачные вычисления, распределенные хранилища.*

Rostyslav P. Strubytskyi *(National University "Lviv Polytechnics", Ukraine)*
Natalia B. Shakhovska *(National University "Lviv Polytechnics", Ukraine)*

ANALYSIS OF APPROACHES TO MODELING OF CLOUD DATA WAREHOUSES

The article carries out the analysis of approaches to deploying the contemporary cloud data warehouses. Key directions in realization of distributed cloud data warehouses are determined along with the major difficulties accompanying. Approaches to increasing the service quality of such warehouses are offered.
Keywords: *data warehouse; cloud computing; distributed warehouses.*

Постановка проблеми. На сьогодні одним із ключових напрямків розвитку інформаційних технологій є хмаркові обчислення. Існує багато визначень хмаркових обчислень, але найбільш вдалим є визначення, запропоноване National Institute of Standards and Technology. Cloud computing – це модель надання повсюдного і зручного мережевого доступу за необхідністю до загального пулу конфігурованих обчислювальних ресурсів (мереж, серверів, систем зберігання, додатків і сервісів), які можуть бути надані і звільнені з мінімальними зусиллями для управління і необхідності взаємодії з провайдером [6].

* статтю підготовлено на основі доповіді на XII-му міжнародному науковому семінарі «Сучасні проблеми інформатики в управлінні, економіці, освіті та екології» (1–5 липня 2013 р., оз. Світязь – Київ).

Аналіз останніх досліджень і публікацій. Концепція «хмаркових технологій» зародилася, коли Дж. Маккарті на конференції MIT Centennial в 1961 р. висловив припущення, що коли-небудь комп'ютерні обчислення будуть виконуватися за допомогою «загальнонародних утиліт» [7]. Ідеологія хмаркових обчислень отримала популярність після 2007 р. завдяки швидкому развитку каналів зв'язку і зростаючих в геометричній прогресії потреб як бізнесу, так і приватних користувачів у горизонтальному масштабуванні своїх інформаційних систем [1].

Хмаркові обчислення пропонують базові сервіси: програмне забезпечення як послуга (SaaS), платформа як послуга (PaaS), інфраструктура як послуга (IaaS). Виділяють кілька моделей розгортання хмарок: приватна, спільноти, спільна і гібридна [9].

Хмарки сьогодні дуже популярні, очевидно, що й надалі вони будуть швидко поширюватися, проте деякі аналітики вважають, що відсутність «всеосяжних» стандартів, незважаючи на те, що зараз над ними працює багато груп, може перетворитися на серйозну перешкоду [10].

Відсутність стандартів є значною перешкодою у використанні хмар, крім того, це може привести до неузгодженості в таких сферах, як безпека і інтероперабельність. Інтероперабельность між додатками і переміщення сервісів при переході від одного провайдера до іншого вкрай важливі для того, щоб споживач міг отримати максимальну віддачу від інвестицій у хмари. Більше того, інтероперабельність дозволить користувачу уникнути необхідності залишатися «заручником» одного провайдера хмарки.

Невирішені частини проблеми. Хмаркове сховище даних (cloud storage) – модель онлайн-сховища, в якому дані зберігаються на численних, розподілених у мережі серверах, що надаються в користування клієнтам в основному третьою стороною [2]. На противагу моделі зберігання даних на власних виділених серверах, кількість і внутрішня структура серверів хмаркового сховища для клієнта, в загальному випадку, невідомі. Дані зберігаються, а також і обробляються в так званій хмарці, яка з точки зору клієнта – один великий віртуальний сервер. Фізично такі сервери можуть розташовуватися віддалено один від одного географічно, аж до розташування на різних континентах.

Широке впровадження хмаркових сховищ даних і необхідність забезпечення високої якості надання послуг вимагає від їхніх розробників цілеспрямованого постійного удосконалення. Під якістю надання послуг розуміють як можливість зберігання даних, так і можливість збереження та отримання доступу до даних за прийнятний час. Для забезпечення цього виникає необхідність у моделюванні архітектури хмаркового сховища даних і процесів, які впливають на якість обслуговування.

Основною проблемою при реалізації і розгортанні сервісів хмаркового збереження даних є географічна віддаленість клієнтів від фізичних реалізацій хмарок [3]. Через це обмін даними між ними відносно повільний.

Метою дослідження є аналіз архітектурних рішень хмаркових сховищ даних і розроблення підходу до визначення оптимального сателіта.

Основні результати дослідження. На практиці вже запропоновані рішення, які передбачають фізичне розміщення файл-сервера на середині відстані між

клієнтами хмарки, що дає можливість урівноважити швидкості завантаження даних на сервер і скачування цих даних з нього. Але такий підхід урівноважує лише можливості клієнтів і не дає змогу вирішити проблему в цілому. Окрім того, даний підхід не дозволяє оптимізувати хмаркове сховище даних для проблем мобільних клієнтів [4].

Історично були запропоновані рішення дзеркальних серверів для збільшення швидкості доступу до файлів. Тобто в хмарці використовуються сервери для реплікації даних один одного. Недоліком такого архітектурного рішення є актуальність даних, що породжує ще одну проблему, не вирішивши попередню [5; 11]. Однією з проблем такого підходу є те, що клієнти повинні самі визначати, на який сервер вони завантажать дані та з якого будуть їх забирати, що обмежує можливості використання даної послуги. Крім того, реплікація даних відбувається тими ж каналами і тими ж протоколами, якими клієнти могли передавати дані безпосередньо.

Існує й інший підхід до вирішення питань розміщення серверів зберігання даних у хмарках. Це так званий підхід мережі доставки і дистрибуції контенту (Content Delivery Network або Content Distribution Network – CDN) – географічно розподілена мережева інфраструктура, що дозволяє оптимізувати доставку та дистрибуцію сервісів кінцевим користувачам в мережі Інтернет. Використання сервіс-провайдерами CDN сприяє збільшенню швидкості завантаження користувачами мультимедійних та інших даних у точках доступу до мережі CDN.

На швидкість обміну даними між користувачем і хмарковим сховищем впливає те, наскільки далеко користувач знаходиться від сервера (відстань вимірюється в мережевих термінах, а не географічних). Це відбувається через те, що при використанні технології TCP/IP, яка застосовується для поширення інформації в мережі Інтернет, затримки при передачі інформації залежать від кількості маршрутизаторів, що знаходяться на шляху між джерелом і споживачем. Розміщення контенту між декількома серверами засобами CDN скорочує мережевий маршрут передачі даних і робить завантаження даних швидшим з точки зору користувача.

Використання CDN знижує кількість хопів, що істотно збільшує швидкість обміну даними між користувачем і хмарковим сховищем. Хоп – назва процесу передачі мережевого пакету між вузлами мережі. Зазвичай, саме хопи використовуються для визначення «відстані» між вузлами. Тобто при зменшенні кількості хопів кінцеві користувачі відчувають меншу затримку при обміні даними, відсутні різкі зміни швидкості завантаження та висока якість потоку даних. Стабільність, яка виникає при цьому, дає змогу доставляти відеодані у форматі HD, забезпечувати швидке завантаження файлів великих розмірів або організовувати відеотрансляцію з високою якістю сервісу (QoS) і низькими витратами на мережу.

Технологія CDN дає змогу уникнути затримок при передачі даних, запобігти можливим перериванням зв'язку і втратам на перевантажених каналах і контактах між ними. Управління навантаженням при передачі мережевого трафіку дозволяє розвантажити магістраль і вузли мережі, розподіливши навантаження між віддаленими серверами.

Розміщення серверів у безпосередній близькості від кінцевих користувачів може значно збільшити вихідну пропускну здатність усієї системи за рахунок розподілення даних на різних серверах.

Сучасні мережі доставки та дистрибуції контента здатні здійснювати автоматичний контроль цілісності даних на кожному з серверів мережі. При цьому гарантується 100% доступність контенту для кінцевого користувача в разі втрати зв'язності між вузлами мережі, виходу з ладу центрального або віддаленого сервера.

Мережі доставки і дистрибуції контента складаються з географічно розподілених багатофункціональних платформ, взаємодія яких дозволяє максимально ефективно обробляти і задовольняти вимоги користувачів.

На сьогоднішній день існує два підходи до реалізації CDN. При першому підході дані центрального сервера Інтернет-ресурсу реплікуються на периферійні платформи. Кожна платформа підтримує в актуальному стані повну або часткову копію розповсюджуваних даних. В іншому випадку дані кешуються на сателітах і зберігаться там деякий час.

Вузол мережі, що входить до складу платформи, взаємодіє з локальними мережами Інтернет-провайдерів і поширює контент кінцевим користувачам найкоротшим мережевим маршрутом з оптимального за завантаженістю сервера. Довжина мережевого маршруту залежить від географічної або топологічної віддаленості користувача комп'ютера від сервера або вартості передачі трафіку в регіоні присутності.

Великі CDN можуть складатися з величезної кількості розподілених вузлів і розміщувати свої сервери безпосередньо у мережі кожного локального провайдера. Багато сервіс-провайдерів сховищ даних роблять акцент на пропускній здатності сполучних каналів і мінімальній кількості точок приєднання. Але, незалежно від архітектури, яка використовується, головним призначенням таких мереж є прискорення передачі як статичного контенту, так і безперервного потоку даних. При такому підході на ключових територіях (зонах) (там, де багато клієнтів) розміщують сервери сателіти, які не зберігають дані та файли, а лише надають швидкий доступ через себе на основні сервери.

Для пришвидшення обслуговування клієнтів і додаткового захисту створюються проміжні сервери, час доступу до яких для клієнтів є меншим, а отже, збільшується їх швидкодія. Такі сервери називають сателітами (проміжні сервери, на яких немає файлів клієнтів). Вони створені для оптимізації передачі даних від основного сервера до клієнта і навпаки. Кількість серверів-сателітів набагато перевищує кількість основних серверів. З урахування вищезазначеного можна представити модель такого сховища у вигляді ієрархії (рис. 1).

Усі сервери розміщені у певних IP-зонах, щоб забезпечити мінімальний час надання послуг клієнтам. Клієнти можуть знаходитися у будь-якій IP-зоні. Крім того, клієнти можуть бути мобільними (змінювати свої IP адреси, переміщатися із зони в зону). Остання обставина вимагає динамічності у визначенні часу доступу до їхньої інформації.

Таким чином, клієнт усю взаємодію з розміщення й отримання інформації проводить безпосередньо через сателіт, незалежно від того, на якому основ-

ному сервері у нього визначене місце для інформації. Звичайно, якщо клієнт статичний, то для нього визначається сателіт, який найближче розміщений до нього. Але ситуація зміниться при переміщення клієнта в іншу зону, технічних неполадках в мережі або за дуже великого навантаження на сателіт від різних клієнтів. При цьому втрачається перевага даного підходу.

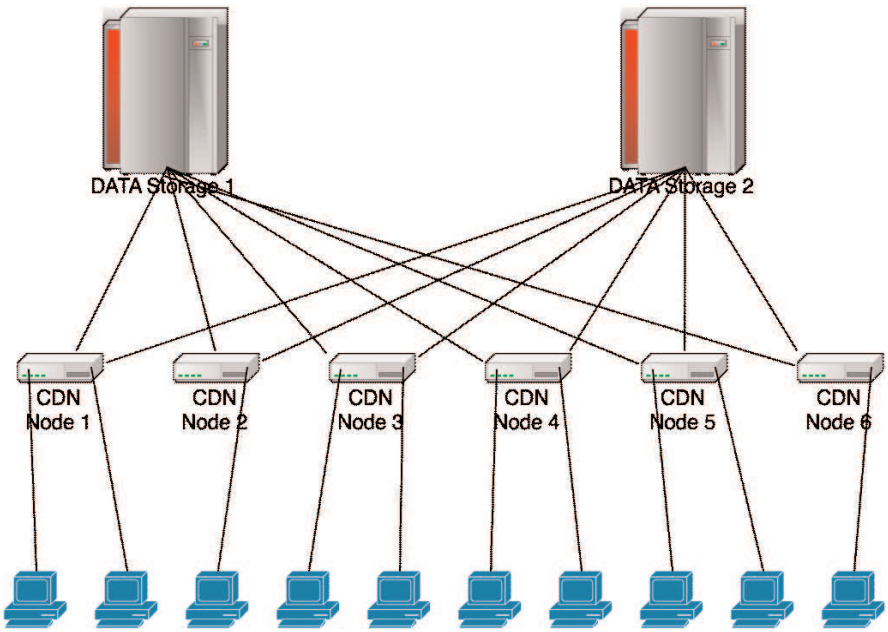

Рис. 1. **Модель ієрархії хмаркового сховища даних,** *авторська розробка*

Для вирішення цієї проблеми необхідно при авторизації клієнта до одного із серверів визначити, який саме сателіт знаходиться до нього найближче.

При надходженні запиту від клієнта до сателіта сервер отримує такі дані про клієнта:

1. IP-адреса клієнта (унікальний ідентифікатор клієнта).
2. RTT (Round-trip delay time) – час від моменту посилання запиту до моменту отримання відповіді.
3. BGP-PATH – шлях прохоження запиту (список провайдерів, через які проходить запит).

Сервер передає ці дані на програмну реалізацію сателіта.

У програмній реалізації клієнта на основі отриманих даних можна отримати шляхи від користувача і кожного із сателітів до клієнта (BGP-PATH), тобто отримати список усіх можливих шляхів від клієнта до потрібної інформації Також, маючи GEOIP, можна визначити географічне місце знаходження клієнта.

Запропонована модель такого вирішення зображена діаграмою потоків даних (рис. 2).

Але отримати RTT від інших серверів до клієнта можна лише в тому випадку, якщо клієнт заходив коли-небудь на них і ця інформація була збережена. Тому потрібно вести статистику відвідування клієнтів з такими даними:

1. IP-адрес клієнта.
2. IP-адрес сателіта.
3. BGP-PATH.
4. RTT.

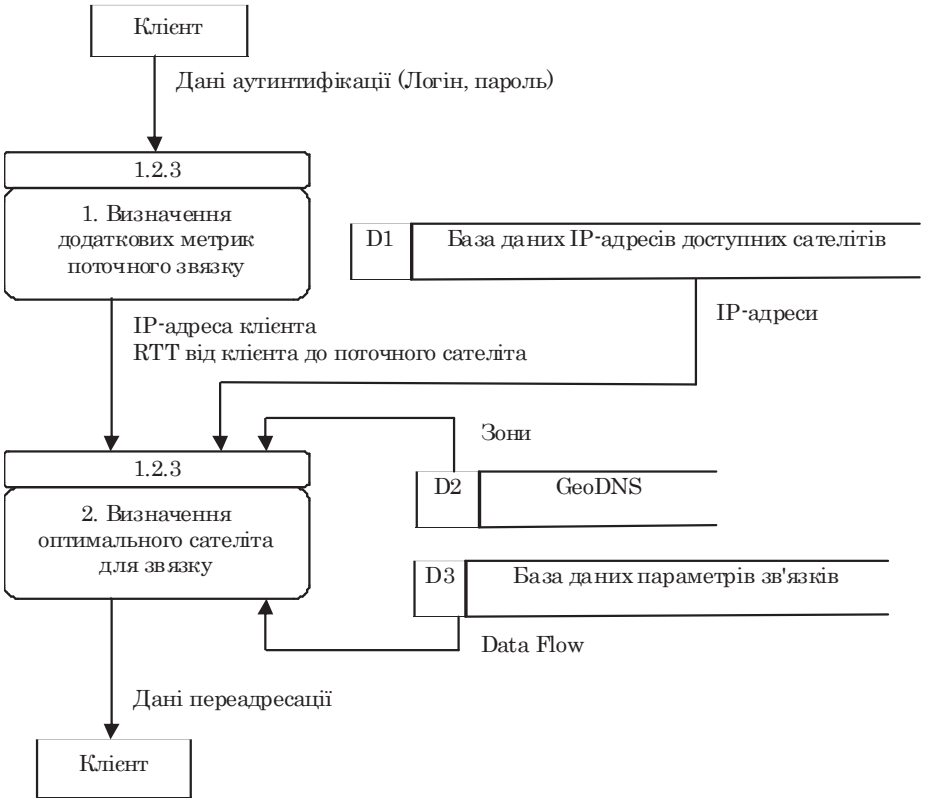

Рис. 2. **Діаграма потоків даних визначення оптимального сателіта,** *авторська розробка*

Крім того, потрібно враховувати завантаженість сателітів на відповідний час роботи з клієнтами.

За відсутності стастичних даних, на основі яких можна прийняти рішення про оптимальний шлях від клієнта до сателіта, можна спробувати оцінити цей шлях за іншою доступною інформацією. Якщо клієнт знаходиться досить близько (GEOIP) до іншого клієнта та їхні BGP-PATH (до одного конкретного сателіта) досить наближені, то можна вважати, що їхній RTT до цього сателіта рівний.

Маючи ці дані, потрібно визначити мінімальний RTT від клієнта до сателіта. На основі цього можна підключити клієнта до його інформаційного сховища через мінімальний шлях, перенаправивши його на ближчий і незавантажений на даний момент сателіт.

Висновки і перспективи подальших досліджень. У результаті аналізу та дослідження різних підходів до практичного розгортання хмаркових сховищ даних було визначено основні слабкі сторони їх реалізації. Визначено шляхи

підвищення якості надання послуг зі зберігання даних шляхом підвищення швидкості отримання доступу до даних. Для подальшого дослідженнями передбачається вивчити впливи різних протоколів передачі даних на якість надання послуг.

Подальші дослідження слід зосередити на перспективному підході з комбінації різних транспортних протоколів на різних ділянках мережевих з'єднань, який можна застосувати на існуючих структурах мережі без їх зміни.

1. Облачные вычисления // www.tadviser.ru.
2. *Риз Дж.* Облачные вычисления (Cloud Application Architectures). – СПб.: БХВ-Петербург, 2011. – 288 с.
3. *Шаховська Н.Б.* Програмне та алгоритмічне забезпечення сховищ та просторів даних: Монографія / Міністерство освіти і науки України, Національний університет «Львівська політехніка». – Львів, 2010. – 194 с.
4. *Antonopoulos, N., Gillam, L.* (2010). Cloud Computing: Principles, Systems and Applications. Springer. 414 p.
5. *Babcock, C.* (2011). IEEE Targets Cloud Interoperability Standards // www.informationweek.com.
6. *Evelyn, B.* (2011). Final Version of NIST Cloud Computing Definition Published // www.nist.gov.
7. John McCarthy (computer scientist) // en.wikipedia.org.
8. *Lawton, G.* (2009). Addressing the Challenge of Cloud-Computing Interoperability // www.computer.org.
9. Mell, P., Grance, T. (2009). The NIST Definition of Cloud Computing // csrc.nist.gov.
10. *Ortiz, S.* (2011). The Problem with Cloud-Computing Standardization. IEEE Computer, 44(7): 13–16.
11. *Shakhovska, N., Medykovsky, M., Stakhiv, P.* (2013). Application of algorithms of classification for uncertainty reduction. Przeglad Elektrotechniczny, 89(4): 284–286.

Стаття надійшла до редакції 18.07.2013.