

УДК 004.043



О.Ф. Лановий, А.К. Кульмінський

ХНУРЕ, м. Харків, Україна, oleksiy.lanovyy@nure.ua

ХНУРЕ, м. Харків, Україна, kulminsky@gmail.com

ВИКОРИСТАННЯ ДАНИХ ЯК СЕРВІСУ ЗА ДОПОМОГОЮ ХМАРНИХ ТЕХНОЛОГІЙ

Описані перспективні принципи агрегування та зберігання даних за допомогою хмарних технологій із використанням принципів Data as a Service(DaaS) та Big Data as a Service(BDaaS)

DaaS, BDaaS, ХМАРА, БАЗА ДАНИХ, ВЕБ-ПРИЛАД, СЕРВЕР

Лановой А.Ф., Кульминский А.К. Использование данных как сервиса при помощи облачных технологий.

Описаны перспективные принципы агрегирования и хранения данных при помощи облачных технологий с использованием принципов Data as a Service(DaaS) та Big Data as a Service(BDaaS)

DaaS, BDaaS, ОБЛАКО, БАЗА ДАННЫХ, ВЕБ-ПРИЛОЖЕНИЕ, СЕРВЕР

Lanovii A.F., Kulminsky A.K. Using data as a service by clouds technology. The perspective principles of aggregation and data storage with the help of cloud technologies are described with the use of the principles of Data as a Service (DaaS) and the Big Data as a Service (BDaaS)

DaaS, BDaaS, CLOUD, DATABASE, WEB APPLICATION, SERVER

Вступ

У теперішній час одним з ключових напрямків розвитку інформаційних технологій є використання інформації в якості сервісу за допомогою хмарних технологій на основі принципів DaaS та BDaaS. Експерти Gartner вважають, що прагнення користувачів ділитися своїми даними і мати до них доступ з різних цифрових пристроїв приведе до того, що до 2020 року половина всіх даних буде зберігатися в «хмарах».

У 2011 році, за підрахунками аналітиків, тільки 7% даних зберігалось на «хмарних» сервісах, а до 2018 року ця цифра має зрости до 36%. За прогнозами Gartner, якщо в 2011 користувачі зберігали приблизно 329 екзабайт даних, то до 2018 року ця цифра збільшиться до 7,3 зеттабайт. Основну частку зазначених обсягів, зазвичай, представляють мультимедійні файли і аналітичні копії існуючих серверів пошукових машин [4].

Суть хмарних технологій, таким чином, полягає в перенесенні обробки даних з персональних комп'ютерів і робочих станцій на сервери всесвітньої мережі. В області комп'ютерного моделювання це означає розгортання програмних комплексів на ресурсах Інтернет. Користувач стає не покупцем обчислювальних програм і комплексів, а їх орендарем, якому надаються різноманітні послуги. Форма купівлі—продажу товару з відчуженням прав власності від продавця до покупця змінюється на форму оренди, в даному випадку — продажу не продукту, а послуг з його використання клієнтом без зміни власника продукту. При цьому забезпечена повна відповідність виробничих потужностей інфраструктури фактичним потребам користувача. Хоча термін **“хмарні технології”** є сталим, в українській мові він має інше значення, ніж оригінал. **“Cloud”** окрім хмари має й інше значення — розсіяний; власне значення **“розсіяний”** і мається на увазі в англійській термінології [3].

Відмінності хмар від традиційних хостинг-рішень

Розглянемо основні відмінності хмар від традиційних хостинг-рішень:

- на відміну від dedicated-серверів, установка і налаштування яких займає багато часу, хмарні сервіси повинні бути доступні для використання відразу після покупки, те ж саме стосується більшості класичних послуг дата-центрів;
- на відміну від shared-хостингу, в хмарах є можливість нарощувати обсяг закуплених потужностей миттєво, без звернення до служби технічної підтримки оператора;
- на відміну від того, що було на ринку хостингу до хмар, хмарні продукти надають схему оплати за фактом, тобто оплату тільки тієї потужності, яку користувач дійсно використовує з досить коротким проміжком тарифікації.

Це основні принципи, за якими можна чітко відрізнити хмарні продукти. Однак, вони досить широкі, хоч і строгі. У поняття хмарного продукту вкладається три види поставки продукту користувачеві за ступенем гнучкості. Це надання інфраструктури як сервісу (в оренду), платформи як сервісу або програми як сервісу.

Характеристики великих даних (Big Data) даних

Згідно з принципом IBM 4V, великі дані мають такі характеристики [1]:

- **Розмір:** розмір даних дуже великий (вимірюється терабайтами і петабайтами), 90% сьогоденних даних було створено всього за останні 2 роки.
- **Різнорозмірність:** різнорозмірність цих даних сягає між структурованими (наприклад, файлами, базами даних), неструктурованими (наприклад, текстові) і напівструктурованими (наприклад, XML), 90% даних неструктурованих, це включає в себе твітів, фондових показників, історію придбання покупців та обслуговування дзвінків клієнтів.

- **Швидкість:** швидкість передачі (наприклад, швидкість твітів відрізняється від швидкості віддалених сканерів чи датчиків для кліматичних змін), 50.000 Гб за секунду є оціночною швидкістю глобального Інтернету трафік до 2018 року.

- **Вертикальність:** визначеність даних, 2 з 3 керівників підприємств мають довіряти інформацію, яку вони використовують для прийняття рішень.

Огляд хмарних сервісів

На ринку представлено кілька типів хмарних сервісів зберігання, призначених для користувача даних:

1. Сервіси, в яких користувачеві надається екземпляр СУБД на віртуальній машині.

2. Сервіси, в яких користувачеві надається СУБД без прив'язки до віртуальних машин і управління віртуальними машинами, масштабованість і відмовостійкість зберігання даних забезпечується постачальником сервісу.

3. Сервіси, в яких користувачеві надається можливість зберігання даних у вигляді таблиць, написання простих додатків для обробки цих даних деякою спеціалізованою мовою.

4. Сервіси зберігання файлів.

В контексті створення хмарного сервісу зберігання даних користувачів за допомогою DaaS-приладів інтерес викликає друга група, як найбільш повно реалізує переваги від застосування хмарних технологій і максимально зручна для користувача, що пов'язано з розробкою спеціального ПЗ для збірки різноманітних джерел даних за запитом клієнтів. Це дозволяє усунювати дозвілні дані за рахунок програмної інтеграції і запобігти невинуватим надмірності даних в мережі на основі сервіс-орієнтованої архітектури та відповідних протоколів. Інтеграція даних підтримує операції з композицією, агрегацією і тимчасовим (віртуальним) поданням даних, що зберігаються у різних власників. Дані залишаються під контролем власника і витягуються на вимогу клієнтів для інтегрованого доступу, можливо на комерційній або вільній основі.

У той же час клієнт — споживач інтегрованої інформації за рахунок єдиного для всіх потреб інтерфейсу позбавлений необхідності витягувати і погоджувати дані від різних джерел в різному форматі, що суттєво економить час і ресурси на розробку свого оригінального програмного забезпечення. При цьому він може і не знати в якій системі, де і як зберігаються цікавлять його дані: реляційна СУБД, файлові ресурси постачальників, результат роботи пошукових сервісів, фіксовані провайдери довідкових даних та ін. Усю роботу з пошуку і перетворення даних можна автоматизувати в рамках єдиної системи зберігання та доступу. Рішення про найкращий спосіб створення цього уніфікованого подання часто вибирається розробниками локальних систем з урахуванням доступності інструментарію, досвіду, кваліфікації та рівня інформаційної

культури організації. При використанні традиційних архітектур час, зусилля і витрати, пов'язані з інтеграцією, можуть звести нанівець її перевага для бізнесу. При реалізації в сервіс-орієнтованій середовищі підхід, заснований на використанні інформаційних сервісів на основі об'єднання у федерацію загальних даних, може істотно поліпшити характеристики повторного використання і знизити його вартість.

Концепції хмарного сховища

Загальноновизнаним стандартом представлення даних на транспортному рівні є XML. Якщо на транспортному рівні використовувати XML-подання, то опис структури збережених даних стає синонімом файлу визначення синтаксису XML-файла DTD ([http:// ru.wikipedia.org/wiki/DTD](http://ru.wikipedia.org/wiki/DTD)) або XML Schema ([Http://ru.wikipedia.org/wiki/XML_Schema](http://ru.wikipedia.org/wiki/XML_Schema)). XML-схеми витіснили DTD в силу їх великих можливостей і універсальності. В цьому випадку схеми подання інформації в XML файлі стає деяким аналогом таблиці в реляційній СУБД або класу об'єктів в об'єктно орієнтованих базах даних.

Таким чином, на основі підходу з використанням проміжного подання даних, що передаються в форматі документів на мові XML можна побудувати сховище даних, які делегуються в нього окремими власниками цих даних як на вільній, так і на комерційній основі. У ролі мови маніпулювання даних в цьому випадку буде виступати мова XQuery, дозволяє виконувати запити до наборів даних, представлених як документами XML, так і традиційними таблицями СУБД. На відміну від мови SQL мову XQuery дозволяє створювати запити до гетерогенним джерел даних, що робить його хорошим інструментом для роботи з розподіленими сховищами, наданими різними провайдером, що входять у федерацію.

У такого сховища буде єдиний інтерфейс для маніпулювання сервісними даними, в тому числі і для дистанційного поповнення з подальшим модеруванням нових даних власниками фізичних копій для забезпечення їх цілісності і коректності. Для розробки дослідного зразка програмного забезпечення в якості полігону можна використовувати публікуються в мережі дані про банках, страхових компаніях, класифікатори адрес і інші джерела колективних даних. Як правило такі дані змінюються не часто і вибірки з них можуть кешуватися на локальних вузлах клієнтів для повторного використання. Оновлення кеша виконується адміністратором за запитом або регулярно за розкладом.

Якщо орієнтуватися на створення універсальної функціонально повної системи зберігання структурованих даних з використанням віртуальних схем на основі загальноприйнятої ідеології використання XML на транспортному рівні, то для такого уявлення потрібно розробити мову і систему інтерпретації для створення довільної схеми даних і її використання при передачі даних між

абонентами корпоративного або федерального хмари. Функціональна повнота системи передбачає наявність засобів додавання і модифікації документів авторизованими віддаленими клієнтами сховища. Тут виникають проблеми з обчислюються атрибутами документів, для яких в загальному випадку не існує алгоритму правильної декомпозиції на складові їх значення полів первинних записів баз даних, розташованих, можливо, в різних вузлах мережі.

Нетривіальною видається завдання підтримки транзакцій та рівнів ізоляції в такий розподіленій системі зберігання даних для формування XML-документів при їх додаванні і модифікації.

Проблема адміністрування сховищ передбачає дослідження стандартних завдань розгортання, міграції, розмежування повноважень, резервного копіювання та відновлення, балансування завантаження, підтримання живучості на достатньому для клієнтів рівні готовності системи за рахунок реплікації копій даних, добре відомих в хмарних технологіях. Їх опрацювання істотно залежить від орендованої інфраструктури хмари і функціональних можливостей наданих провайдером віртуальних машин і їх програмного забезпечення в рамках DaaS чи BDaaS. Для транспортування даних між клієнтами і серверами сервісного сховища можна використовувати протоколи передачі на основі SOAP і REST.

На рис. 1 наведена укрупнена схема, що ілюструє проходження запиту в корпоративне сховище.

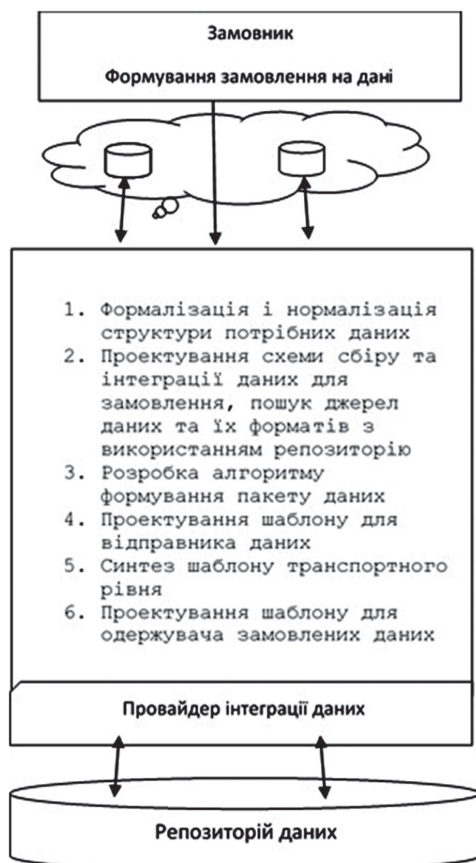


Рис. 1. Схема функціонування сховища технології DaaS

Формування замовлення на передачу даних одержувачем передбачає роботу з редактором, що дозволяє сформулювати вимоги до запитуваною даними з боку користувача у вигляді шаблонів, що містять перелік даних та їх формат. Вимоги формуються на основі метаданих про всій номенклатурі даних хмарних сервісів, що зберігаються в репозиторії системи.

Користувачеві повинні бути подані шаблони запитів з сховища, раніше створені іншим користувачами, з можливістю їх адаптації під конкретні вимоги.

Формалізація і нормалізація структури затребуваних даних передбачає побудову формально-логічної схеми запитаних даних, контроль їх доступності, форматів зберігання і форматів подання до результатуючих наборів.

Проектування схеми збору / інтеграції даних для замовлення, пошук джерел даних і їх форматів з використанням сховища передбачає формування гетерогенних запитів до власників даних, отримання і перекодування даних відповідно до вимоги результатуючої схеми.

Розробка алгоритму формування пакета даних передбачає автоматизоване створення програмної реалізації виконання спроектованих раніше запитів, отримання та формування даних у відповідно до побудованими схемами, формування вихідних наборів даних для передачі користувачу.

Проектування шаблону для відправника даних передбачає створення XML Schema для відправки даних по мережі замовнику.

Синтез шаблону транспортного рівня передбачає запаковування даних з використанням існуючих доступних протоколів передачі даних.

Проектування шаблону для одержувача затребуваних даних передбачає розпаковування даних протоколу транспортного рівня і передачі їх замовнику.

Репозиторій системи є базою даних, в якій зберігаються відомості про всієї доступної інформації в системі. За кожною одиницею зберігання (ресурсу) в репозиторії повинна бути інформація про сенс (Семантиці), представленою метаданими, локалізації джерела, типі сховища, структурі таблиці або документа, типі, форматі зберігання.

Типові хмарні архітектури

Традиційні архітектури хмарних систем передбачають їх побудову на базі багаторівневої мережевої інфраструктури віртуальних центрів обробки даних (ЦОД) [5] і підтримки у вигляді програмної платформи з набором сервісів. Типова архітектура будь-якого хмарного ЦОД має ієрархічну структуру, що складається з 3 основних рівнів:

Рівня ядра (Core). На цьому рівні працюють маршрутизатори або комутатори 3 рівня, які складають основу всієї мережі дата-центру з високошвидкісними портами (10/40/100 GbE) для маршрутизації потоків між WAN і мережею дата-центру.

Рівня агрегації (Aggregation або Distribution). На цьому рівні також функціонують комутатори 3 рівня, основне призначення яких – розподіл навантаження між локальними мережами дата-центру.

Рівня доступу (Access). На цьому рівні розташовуються кінцеві точки (сервера, робочі станції та ін.) і мережеве обладнання, що зв'язує кінцеві точки з рівнем агрегації. На рівні доступу функціонують кластера дата-центрів, що складаються з множини фізичних серверів і великої кількості віртуальних машин на кожному з них. На цьому ж рівні найчастіше розташовується загальна мережа зберігання даних SAN (Storage Area Network). Група, що складається з взаємозв'язаних компонентів зберігання даних, обчислювальних і мережевих ресурсів, які працюють спільно на рівні доступу, щоб надати сервіс або додаток клієнту називається точкою доставки або POD (Point of delivery).

Висока готовність віртуального ЦОД досягається за рахунок дублювання мережних пристроїв на кожному рівні та використання резервних маршрутів та методів балансування навантаження.

Компоненти найнижчого рівня – сервера, які об'єднуються в кластера. На кожному фізичному сервері функціонує декілька віртуальних машин, тобто на одному фізичному мережному інтерфейсі є декілька віртуальних, обміном між якими управляє програмний компонент гіпервізор (Virtual Machine Monitor /Hypervisor).

Хмарні IT-інфраструктури навчального закладу включають в себе наступні компоненти:

- ядро хмари – ядро всієї IT-інфраструктури;
- персональні комп'ютери в лабораторіях, які є вузлами хмари.

Можливості мов програмування

Найбільш загальні можливості мають засоби мов програмування, наприклад, Java для обробки XML-документів [5]. Java-розробнику для створення XML-додатків доступні не менш шести розширень Java-платформи:

- Java API для обробки XML (JAXP – Java API for XML Processing);
- Java API для зв'язку XML з Java (JAXB – Java API for XML / Java Binding);
- Довгострокова персистенція JavaBean-компонентів;
- Java API для обміну повідомленнями XML (JAXM – (JAXM – Java API for XML Messaging);
- Java API для RPC XML (JAX RPC – Java API for XML RPC);
- Java API для реєстру XML (JAXR – Java API for XML Registry).

Базові інструменти для роботи з документами JAXP надає розробнику Java і технології для обробки XML документів, які залежать від JAXP. Серед них:

- SAX – простий API для XML;
- DOM – програмний інтерфейс API об'єктної моделі документа (Document Object Model) від консорціуму W3C;

- XSLT – мова перетворень каскадної таблиці стилів XML від консорціуму W3C;

- XPath – мова XML XPath від консорціуму W3C;

- JDOM – API оптимізованої об'єктної моделі документа від jdom.org.

Веб-сервіси є програми сервера і додатки клієнта, які зв'язуються через протокол World Wide Web (WWW) HyperText Transfer Protocol (HTTP). На концептуальному рівні, сервіс є програмним компонентом, доступним через кінцеву точку мережі (endpoint). Споживач і постачальник сервісу використовують повідомлення, щоб обмінюватися викликами запитів і отримання відповідної інформацією в формі самодостатніх документів, які припускають лише деякі припущення про технологічні можливостях одержувача [1].

На технічному рівні, веб-сервіси можуть бути реалізовані різними способами: JAX-WS веб-сервіси "Big" і JAX-RS веб-сервіси "RESTful".

JAX-WS технологія Java EE забезпечує функціональність для сервісів, що використовують повідомлення Extensible Markup Language (XML), які використовуються стандартом Simple Object Access Protocol (SOAP), визначальним архітектурі і формати повідомлень на мові XML. У таких системах часто опис сервісних операцій пропонується записувати на мові Web Services Description Language (WSDL) – варіанти мови XML для синтаксичного визначення інтерфейсів.

JAX-RS веб-сервіси на базі REST (REpresentational State Transfer – "RESTful") є колекціями сервісів, ідентифікованих своїми URI. Кожен документ або процес моделюється веб-ресурсом зі своїм унікальним URI. Ці веб-ресурси управляються діями, які можуть бути специфіковані в заголовку HTTP. Чи не використовуються ні стандарт SOAP, ні WSDL, ні стандарти WS. Замість них обмін повідомленнями може проводитися в різних форматах: XML, JSON, HTML і ін. Часто клієнтом є веб-браузер. Технологія в силу можливостей використовуваних транспортних протоколів не забезпечує захист транспортується інформації від несанкціонованого доступу, що є важливою особливістю RESTful. RESTful добре підходить для основних і оперативних (на льоту) сценаріїв інтеграції. веб-сервіси RESTful часто краще інтегровані в HTTP, ніж SOAP-орієнтовані сервіси.

Засоби JAX-RS в Java EE забезпечують функціональне призначення для сервісу RESTful. Вони не вимагають XML повідомлень або визначень WSDL сервісу. Проект JAX-RS – промислова готова реалізація для JSR 311: JAX-RS: Java API для RESTful Web Services. JAX-RS здійснює підтримку анотацій, визначених у JSR-311, полегшуючи розробникам формування веб-сервісів RESTful для Java і Java Virtual Machine (JVM).

Для роботи з XML документами консорціумом W3C розроблений мова запитів XQuery [4], [5], спеціально орієнтований на витяг даних зі складних структур документів.

На ринку інформаційних технологій існує широкий набір вільних і комерційних процесорів для обробки XQuery: Zorba (<http://sourceforge.net/projects/zorba/>), Xqilla (<http://xqilla.Sourceforge.net/HomePage>) та інші. Повний список продуктів цієї серії можна подивитися на <http://www.w3.Org/XML/Query/#implementations>. прикладом вільного ПЗ для обробки запитів є Xqilla – процесор XQuery написаний на мові C++ і викликається в інтерпретаторі командного рядка: xqilla [options] <XQuery file>.

Для виконання вимог розробки сервісного сховища хмарний сервіс зберігання даних користувачів DaaS-додатків повинен включати балансувальник навантаження і кілька однакових вузлів центрального сховища з сервером обробки запитів користувачів сховища (рис. 2).

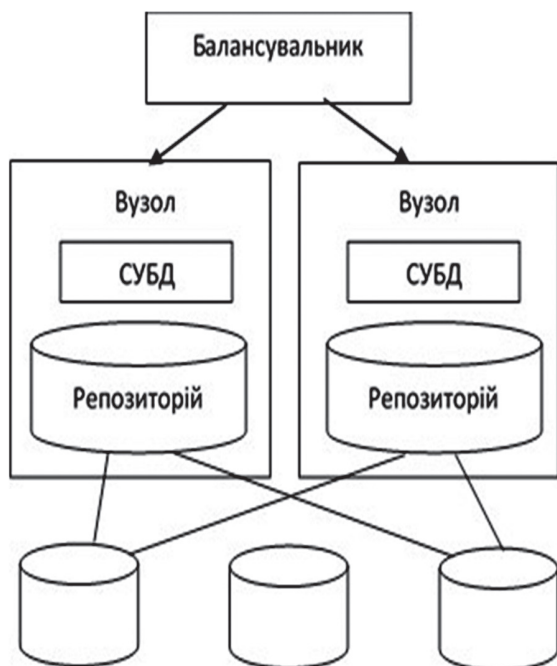


Рис. 2. Архітектура DaaS сховища

Репозиторій містить опис всіх доступних даних сховища, що містить такі відомості:

- внутрішній ідентифікатор запису сховища;
- зовнішнє ім'я блоку даних для ідентифікації користувачами;
- метадані - коментар вмісту блоку даних;
- формату, правил доступу та ін.;
- тип збережених даних: ключ-значення, XML-документ, файл і т.
- адреса зберігання;
- протокол доступу;
- порт доступу;
- Login/Password/рівень доступу користувачів (читання, запис);
- текст запиту але замовчуванням і його параметри;
- XML-схема подання результату запиту;
- місце обробки запиту: локальний вузол користувальницький центральний сервер;
- конвертор результату в XML-документ;
- власник ресурсу;

- ціна одиниці ресурсу;
- дата поновлення ресурсу;
- обсяг ресурсу;
- обробник запиту GET;
- обробник запиту POST;
- інші відомості.

Обробка запиту GET.

У запиті користувача па отримання ресурсу зазначаються:

- ідентифікатор ресурсу в репозиторії сховища;
- характер запиту: за замовчуванням або призначений для користувача;
- параметри для запиту за замовчуванням;
- текст запиту користувача, що включає форму результату.

Результат виконання запиту за замовчуванням завжди у вигляді XML-документа для забезпечення універсальності (прозорості) подання. Призначені для користувача запити форматовуються самостійно.

Залежно від типу ресурсу і СУБД обробка запиту може бути передана на локальний сервер СУБД зберігання ресурсу або виконана на центральному сервері сховища для запитів на мові XQuery до XML-документах локальних сховищ. Для цього ресурс прошу оформити на центральний сервер і може бути кеш, наприклад, в окремому полі записи про ресурс для економії трафіку в майбутніх запитах.

Якщо вміст ресурсу контролюється тільки локальної SQL СУБД, то контроль дати поновлення утруднений / неможливий. Тому кешувати такий ресурс не має сенсу.

Обробка запиту POST.

У запиті користувача на зміну ресурсу зазначаються:

- ідентифікатор ресурсу в репозиторії сховища;
- текст запиту користувача на зміну записів, в залежності від типу СУБД.

Після обробки запиту коригується поле дати зміни і, можливо, копія в кеші.

Переваги хмарних технологій

Перевагами хмарних обчислень є те, що користувач має можливість не купувати потужні комп'ютери. Зокрема, і організації можуть відмовлятися від придбання потужних серверів і йти "в хмари". Для розробника – контрольованість усього процесу. У разі виникнення проблеми їм істотно простіше буде змодельовати ситуацію, що викликала помилку, – адже усі дані і так зберігаються в них. Користувач оплачує послугу тільки тоді коли вона йому потрібна, а найголовніше він платить тільки за те, що використовує.

Хмарні технології дозволяють економити на придбанні, підтримці, модернізації ПЗ і устаткування [5].

Масштабованість, відмовостійкість і безпека – автоматичне виділення і звільнення необхідних ресурсів залежно від потреб додатку. Технічне обслуговування, оновлення ПЗ здійснює провайдер послуг.

Віддалений доступ до даних у хмарі — працювати можна з будь-якої точки на планеті, де є доступ в мережу Інтернет.

Недоліки хмарних технологій

Розглядаючи переваги “хмарних” обчислень, варто сказати і про недоліки, з якими зв’язаний перехід на “хмари”. Найбільш суттєвий з них — загроза інформаційної безпеки. В умовах жорсткої конкуренції, найбільше компанії бояться витоків даних з мережі “хмарного” провайдера внаслідок перехоплення інформації, втрати контролю над даними і додатками, неможливості знищення даних, дій інсайдера на стороні провайдера або інших користувачів “хмари”. Для захисту можна використовувати шифрування даних або їх знеособлення. При цьому шифрувати треба не лише ті дані, що зберігаються в провайдера, а й канал зв’язку з ним. Проте доки рішення, які дозволяють ефективно захищати дані в “хмарі”, не вироблені.

Ще одним недоліком можна назвати прив’язку “хмарної” технології до конкретного постачальника послуг, збоїв на стороні провайдера, вихід з ладу інтерфейсу адміністрування, банкрутство і поглинання оператора. Компанії не даремно побоюються цих подій, оскільки це може принести їх бізнесу значний матеріальний збиток.

До інших ризиків можна віднести втрату зв’язку з мережею провайдера, DDoS-атаки і втрату відповідності вимогам регулювальників. Ці ризики можна понизити за допомогою правильного складання угоди про рівень обслуговування (Service Level Agreement, SLA), яке дозволить компенсувати частину збитків. Нормативні вимоги можуть змінюватися з часом, а закон “Про персональні дані” і зовсім робить “хмарні” обчислення непридатними на практиці. Проте, в деяких випадках хмарну систему можна зробити навіть більш захищеною, ніж традиційну архітектуру, за рахунок розподілу обов’язків і правильно складених домовленостей [4].

В першу чергу, це, звичайно, повна залежність користувача від підключення до Інтернету. Без Інтернету відразу пропадає все — листи, документи, контакти, ігри, заплановані завдання, встановлені будильники тощо. Ну і, звичайно, зростає ціна помилки. Якщо у виробника виникає будь-який збій, він ризикує втратити або видати у відкритий доступ усі дані. Так, наприклад, у 2009 році сервіс для зберігання закладок Magnolia втратив усі свої дані. Не можна сказати, що вірогідність цього вища, ніж поломка або втрата ноутбука користувача, але просто масштаб катастрофи може бути дійсно великий.

Користувач не є власником і не має доступу до внутрішньої хмарної інфраструктури. Збереження призначених для користувача даних залежить від компанії провайдера. Зазначимо певні недоліки:

- недолік актуальний для українських користувачів: для отримання якісних послуг користувачеві необхідно мати надійний і швидкий доступ до мережі Інтернет.
- відсутність загальноприйнятих стандартів у напрямі безпеки хмарних технологій.

Висновок

Роботи в області створення корпоративних сховищ у світовій практиці знаходяться в початковому стані, що дозволяє сподіватися на отримання в цьому напрямку результатів світового рівня, отримати пріоритет в просуванні цієї технології як в науковому плані, так і в плані комерціалізації результатів інтелектуальної діяльності. Цьому сприяє готовність ринку вільних інструментальних засобів розробки веб-додатків і веб-сервісів на мовах програмування Java, Python, Ruby on Rails, широко використовуваних в хмарних технологіях, і наявний широкий досвід використання цих інструментів для розробки веб-технологій для хмар.

Список літератури: 1. The Java EE 6 Tutorial — Oracle [Електронний ресурс]. Режим доступу: <https://docs.oracle.com/javase/6/tutorial/doc/> (дата звернення: 17.08.2017) — Назва з екрана. 2. Построение XML (XQuery). [Електронний ресурс]. Режим доступу: <https://docs.microsoft.com/ru-ru/sql/xquery/xml-construction-xquery> (дата звернення: 10.05.2017) — Назва з екрана. 3. С. М. Саракко, Д. Чамберлін. Р. Ахуджа. DB2 и XML: Запрос XML-данных при помощи Xquery. [Электронный ресурс]. Режим доступу: <http://www.ibm.com/developerworks/ru/library/db2xml-5/index.html> (дата звернення: 30.09.2017) — Назва з екрана. 4. Облачные вычисления (мировой рынок) [Електронний ресурс]. Режим доступу: http://www.tadviser.ru/index.php/Статья:Облачные_вычисления:мировой_рынок (дата звернення: 15.11.2017). 5. Хорстманн К., Корнелл Г. Java 2. Библиотека профессионала, том 2. Расширенные средства программирования. 10-е издание: Пер. с англ. — М.: Диалектика-Вильямс, 2016. — 968 с.

Resume

Lanovii A.F., Kulminsky A.K. USING DATA AS A SERVICE BY CLOUDS TECHNOLOGY

Background: At present, one of the key areas for the development of information technology is the use of information as a service through cloud-based technologies based on the principles of DaaS and BDaaS. There are several types of cloud storage services available for the user on the market, the features of which are described in this article.

Materials and methods: The article analyzed the concepts of cloud technologies and the possibilities of modern programming languages (for example, Java) for the possibility of implementing the principle of data as a service. The schemes of functioning of corresponding service systems are resulted and processing relevant data requests in their environment.

Results: The analysis provides recommendations on the use of modern programming languages to create systems based on the principles of DaaS data storage formats in these systems, methods and features of access to them.

Conclusion: Work on the creation of corporate storage facilities in the world practice resides in the initial state, which allows us to hope for obtaining in this direction world's class results of the world, to receive the priority in advancing this technology both scientifically and in terms of commercialization results of intellectual activity. This is facilitated by the readiness of the market for free tools for developing web applications and web services in the Java programming languages, Python, Ruby on Rails, widely used in cloud-based technologies, and there is a widespread use of these tools for the development of web technologies for clouds.

Надійшла до редакції 06.10.2017