

ЕВОЛЮЦІЯ ПОБУДОВИ АРХІТЕКТУР ІНФОРМАЦІЙНИХ СИСТЕМ. ПЕРСПЕКТИВИ РОЗВИТКУ “ХМАРНОЇ” АРХІТЕКТУРИ

© Бойко Н. І., 2015

Обґрунтовано теоретичні положення, запропоновано методичні та практичні рекомендації для підвищення дієвості функціонування інформаційної системи. Проаналізовано основні моделі побудови інформаційних систем, наведено переваги та недоліки розподілених та централізованих корпоративних інформаційних систем. Обґрунтовано методологію формування інформаційних потоків для їх застосування в “хмарних обчисленнях”. Аналізується сервісно-процесний підхід до організації інформаційних потоків у “хмарі”. Обґрунтовано доцільність використання “хмарних обчислень” для ефективної роботи бізнес-структур.

Ключові слова: система, інформаційний потік, інформаційні процеси, “хмарні обчислення”, корпоративні інформаційні системи, інформаційна технологія.

In the article the theoretical principles are outlined, methodological and practical recommendations to enhance the effectiveness of the information system are proposed. The analysis of the main models of building information systems is carried out. The advantages and disadvantages of distributed and centralized corporate information systems are given. Methodology of forming of information flows for their use in “cloud computing” is grounded. Analyzes the service-process approach to organizing information flow in “the cloud.” The feasibility of using “cloud computing” for effective work of businesses is substantiated.

Key words: system, information flow, information processes, “cloud computing,” corporate information systems, information technology.

Вступ. Загальна постановка проблеми

Через зростання обсягу українського ринку системної інтеграції та трансформації сьогодні існує безліч цікавих пропозицій з використання ІТ-інфраструктури для ефективного управління бізнесом. Широкого використання набули розподілені корпоративні інформаційні системи (КІС), організація роботи яких залежить від безперебійної роботи створених на місцях центрів обробки інформації (ЦОД). Центри призначені для автоматизації процесів збирання та пересилання інформації, її аналізу та створення звітів з використанням економіко-математичних методів, технічних засобів та організаційної структури, яка забезпечує раціональне керування складними об'єктами й інформаційними процесами. За допомогою КІС можна розв'язувати задачі первинного та оперативного планування виробництва, оперативного розподілу завантаження обладнання та розподіленого використання ресурсів організації.

Сьогодні більшість функцій КІС еволюціонувала у функції Business Intelligence Network Ergatic Organism, які дозволяють їм працювати у сучасних мережевих автоматизованих системах управління (АСУ). Для кращої реалізації КІС у мережі слід передбачити можливість передавання інформації для аналізу засобами Business Intelligence у поєднанні з елементами мережевої системи управління знаннями Key Management Service (KMS). Така еволюція АСУ, своєю чергою, призвела до реалізації інформаційно-аналітичних та управлінських можливостей у платформі Cloud computing (“хмарне обчислення”).

Використання “хмарних обчислень” стало важливим фактором для розвитку Business Intelligence, Business Intelligence NEO та Network Ergatic Organism (NEO), який є перспективним

напрямок для використання ІТ-платформ. Хоча перехід у “хмарні обчислення” вирізняється певними протиріччями морально-етичного, юридичного, технічного характеру. Ефективне користування цією технологією має низку позитивних моментів. Серед них можна назвати: високий рівень відмовостійкості, гнучкість конфігурації під потреби певного проекту, економія часу на організацію інфраструктури, зручність та простота у керуванні серверними ресурсами в єдиному інтерфейсі, зміна та докомплектація потужностей без зупинки “хмари” та автоматизація всіх ІТ-процесів. Відомо, що процес автоматизації розпочинається з виділення ресурсної потужності, підтримки відмовостійкості, модернізації платформи до розвитку віртуального дата-центру клієнтів.

При використанні “хмарних обчислень” перспективним для бізнес-моделей є можливість управління витратами на програмне та технічне забезпечення й ефективне його використання, приміром, як послуги Software as a Service (SaaS). Успішні хмарні рішення слід застосовувати як зміну або розширення традиційних програмних рішень для технічної та фундаментальної реорганізації бізнес-моделі. За їх допомогою можна зберегти надвеликі обсяги даних, доступ до яких здійснюється з мобільних та персональних комп’ютерів.

Аналіз останніх досліджень та публікацій

Аналіз досліджень авторів [1, 3, 5, 16] свідчать про те, що величезної актуальності набувають тенденції впровадження в бізнес-структури “хмарних технологій”. Їх використання дає змогу організувати доступ до програмного забезпечення при застосуванні своїх інформаційно-аналітичних завдань. На думку авторів [2, 3], напрямок “хмарних обчислень” є пріоритетним в галузях математичного та інженерного спрямування. Проте широкого використання цих технологій в управлінській роботі бізнес-моделей не спостерігається, що підвищує актуальність цієї проблематики та сприяє розвитку цього наукового напрямку.

Автори [8, 9, 11, 12] розглядають поточний стан та перспективи використання “хмарних технологій” та сервісів на ринку України. У статті проаналізовано роботи [6–8, 10], в яких розглядаються хмарні технології безпеки. Також в роботі [19–21] розглядається архітектура “хмари” та обґрунтовується ефективність роботи її сервісів.

У дослідженні [23] охарактеризовано технологію хмарних сервісів, проаналізовано основні задачі та принципи її інформаційної безпеки, визначено перспективи розвитку як самої технології, так і параметрів її безпеки.

Сьогодні диктує жорсткі умови для діяльності підприємств і деякі з них звертають увагу на “хмарні сервіси”. Застосування сервісів у бізнесі дає реальний шанс підвищити його ефективність та конкурентоспроможність. В умовах економічної нестабільності компанії повинні жорстко контролювати свої витрати, адже якість та цінність будь-якої інновації повинна бути фінансово обґрунтованою. Тому **метою статті** є аналіз інформаційних технологій, робота яких дає змогу підвищити економічний та управлінський ефект від діяльності різних архітектур інформаційних систем. Обґрунтовано доцільність використання “хмарних обчислень” для ефективної діяльності динамічних систем.

Еволюція архітектур інформаційних систем

За весь період еволюції ІТ проіснувало декілька моделей побудови інформаційних систем. Розпочиналось все із монолітної архітектури мейнфреймів, в яких база даних і додатки працювали на одному комп’ютері (рис. 1). Перевагою монолітної архітектури є централізована багатопотокова та багатозадачна обробка всіх інформаційних потоків в ІС. Така організація інформаційних процесів дає змогу оптимізувати одночасну обробку великої кількості процесів. Під час роботи мейнфрейму користувачу ІС та кожному інформаційному процесу виділяється комплекс інформаційних ресурсів для вирішення поставлених завдань. Вони можуть ізолювати та виправляти більшість апаратних та програмних помилок [6].

З розвитком мережових технологій ускладнилась архітектура ІС. Першим видом багатокористувацької системи є файл-серверна організація даних (рис. 2). У цьому виді архітектури

шукають та обробляють інформацію на робочих станціях, де при виконанні запиту користувачу надсилаються не тільки необхідні дані, але й дані, які використовуються виключно для виконання запиту. За такої організації інформаційних потоків виникає надлишок обсягу інформації.

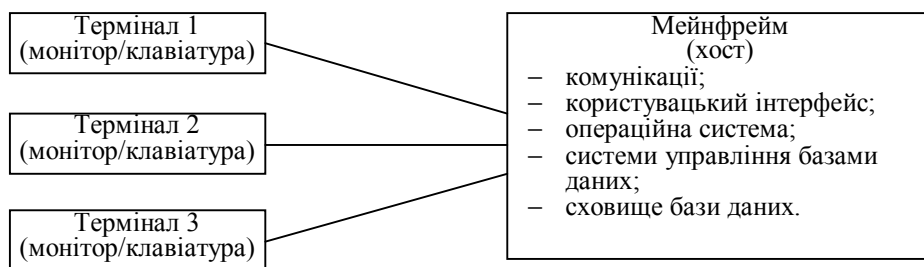


Рис. 1. Архітектура мейнфреймів

За цієї архітектурної організації час на виконання запиту складається з часу передавання даних з файл-сервера на робочу станцію та часу на виконання самого запиту на робочій станції [7–9, 11, 12].

Збільшити швидкодію файл-серверної технології можливо з застосуванням систем управління баз даних (СУБД). Синхронна взаємодія між СУБД різних користувачів та файл-серверною системою приводить до втрат інформації [14].

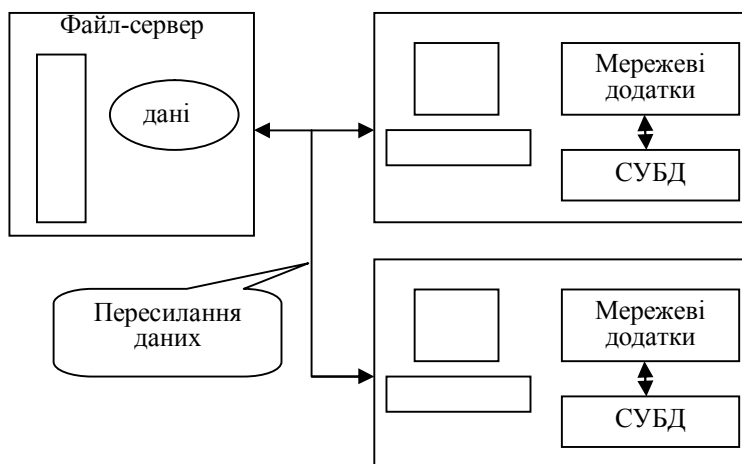


Рис. 2. Схема файл-серверної архітектури

У процесі постійного розвитку комп'ютерних мереж якісно розвивається ідея розподілу ресурсів. У розподілених ІС ефективного доступу до вузлів комп'ютерної мережі можна досягти завдяки високій пропускній здатності локальних мереж. При цьому виділяються основні компоненти мережі, тобто існує не тільки доступ до ресурсів, але й можливість отримувати комплекс сервісів, не дублюючи їх в кожному вузлі мережі. Виконання інформаційних процесів у розподіленій ІС дає змогу робочим станціям працювати автономно і вибирати потрібні їм сервіси мережі [23].

Переважно робочі станції та програми, які входять до складу ІС, не є рівноправними: одні з них мають ресурси (файлова система, процесор, БД тощо), а інші ними користуються. Тому комп'ютер, який управляє ресурсом, називають сервером цього ресурсу, а користувачі, які працюють за цим комп'ютером, називаються клієнтами ІС. Така архітектура отримала назву "клієнт-сервер". Її практична реалізація зветься клієнт-серверними технологіями, в яких прописано власні правила обміну – протоколи взаємодії (протоколи обміну) [1–3].

У будь-якій мережі присутні елементи клієнт-серверної технології, які поділяються на два типи: дворівневу та тривірневу архітектуру. Дворівнева архітектура (two-tier) отримала свою назву через розташування в ній трьох базових компонентів, які з'єднані двома вузлами (рис. 3).

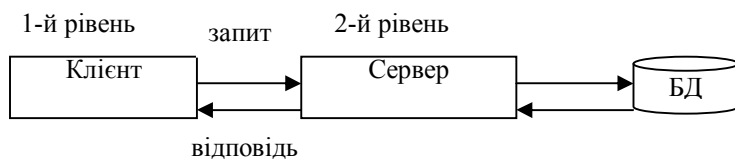


Рис. 3. Дворівнева клієнт-серверна архітектура

У дворівневій архітектурі сервер напряму відповідає на запити клієнта, користуючись лише своїми ресурсами. Тобто сервер не викликає додаткових мережевих додатків та ресурсів для виконання всього запиту чи якоїсь його частини. На рис. 4 наведено модель взаємодії клієнтської та серверної частин з її складовими.

Аналізуючи рис. 4, можна виділити чотири основні моделі взаємодії клієнта та сервера в дворівневій архітектурі [4]:

- термінал-сервер – розподілене представлення даних;
- файл-сервер – доступ до віддаленої БД та файлових ресурсів;
- сервер БД – віддалене представлення даних;
- сервер додатків – віддалені додатки.

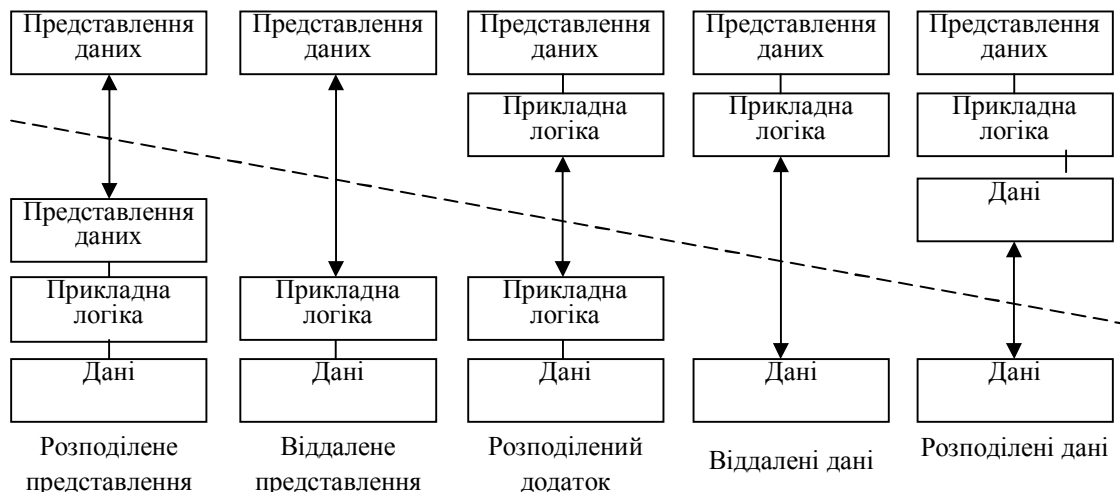


Рис. 4. Моделі клієнт-серверної взаємодії

У процесі розвитку ІС актуальною є централізація обчислень, яка реалізується в моделі термінал-сервер. Термінали містять мінімум програмних та апаратних засобів та представляють мультимедійний доступ до даних у системі. Роботу таким терміналам забезпечує високовиробничий сервер, в якому розташовано потрібні програмні засоби, зокрема віртуальні драйвери апаратних засобів [19].

Використання розподілених обчислень, що реалізуються на основі моделі сервера додатків, призвела до виникнення тривірневої клієнт-серверної архітектури (three-tier). У цьому архітектурному рішенні додатки винесено на окремий сервер додатків, а користувачі працюють через браузер на "тонких" клієнтах (рис. 5). Три рівні клієнт-серверної ІС взаємодіють між собою у заздалегідь узгодженому форматі.

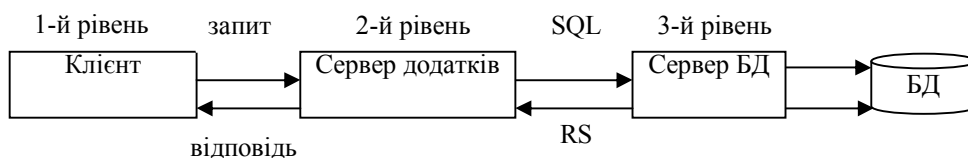


Рис. 5. Трирівнева клієнт-серверна архітектура

На рис. 5 продемонстровано трирівневу клієнт-серверну архітектуру, в якій компоненти розподілено так [20]:

- представлення даних – на боці клієнта;
- прикладний компонент – на виділеному сервері додатків;
- управління ресурсами – на сервері БД, який відображає дані запиту клієнта.

Трирівнева клієнт-серверна архітектура може розширюватись до багаторівневої клієнт-серверної архітектури (N-tier, Multi-tier) (рис. 6), в якій додаткові сервери містять власні сервіси та користуються послугами інших серверів різних рівнів.

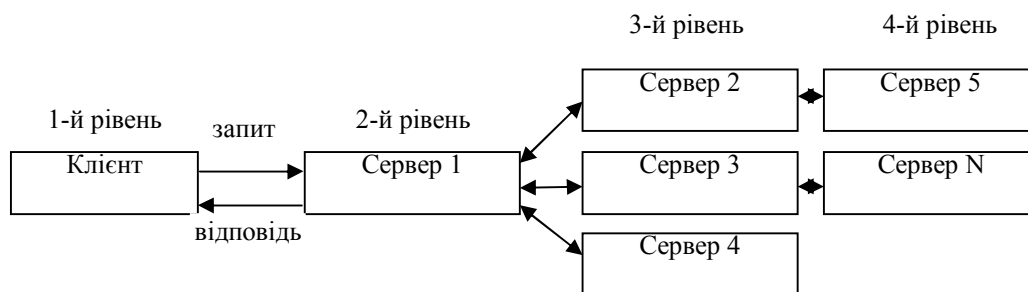


Рис. 6. Абстрактне представлення багаторівневої клієнт-серверної архітектури

Аналізуючи клієнт-серверні технології, можна зробити висновок, що дворівнева клієнт-серверна архітектура є простішою в обслуговуванні запитів одним сервером, але вона не є надійною. Трирівнева архітектура є складною та дає змогу підтримувати високий ступінь гнучкості, масштабованості та забезпечувати безпеку та продуктивність серверів [20].

Клієнт-серверні архітектури застосовуються у мережесих технологіях для доступу до мережесих сервісів [10]:

1. Web-сервери дають доступ до гіпертекстових документів за протоколом HTTP (Hyper Text Transfer Protocol), також до бінарних файлів (зображення, мультимедіа тощо).
2. Сервери додатків призначені для централізованого вирішення прикладних задач у деякій предметній області. Вони спрощують конфігурацію клієнтів та управління мережею.
3. Сервери баз даних використовуються для обробки клієнтських запитів мовою SQL, СУБД.
4. Файл-сервери зберігають інформацію у вигляді файлів та спрощують доступ до неї. Вони є захищеними від несанкціонованого доступу.
5. Проксі-сервер є посередником між кеш-пам'яттю та користувачем в отриманні необхідної інформації.
6. Файрволи (брандмауери) – це міжмережеві екрани, які аналізують та фільтрують мережесий трафік з метою безпеки мережі.
7. Поштові сервери пропонують послуги з відправлення та отримання електронних поштових повідомлень.
8. Сервери віддаленого доступу RAS (Radmin Server) забезпечують зв'язок клієнтів з сервером комутуваними лініями.

Для доступу до мережесих сервісів використовуються клієнти, “товщину” яких визначає конфігурація обладнання та програмне забезпечення. Отож, “тонкий” клієнт може здійснити запуск

необхідного мережевого додатку через web-інтерфейс, а “товстий” – робочу станцію або ПК, який працює під власною дисковою операційною системою, що містить необхідне програмне забезпечення.

Останнім часом стає популярним термін “rich-client”, який поєднує властивості “тонкого” та “товстого” клієнтів. Він, як “тонкий”, має графічний інтерфейс із записами XML та функціональність “товстого” (інтерфейс drag-and-drop, закладки, багатовіконний режим, випадаючі діалогові вікна тощо). На базі XML працюють такі протоколи: XAML (eXtensible Application Markup Language) – розроблений Microsoft та використовується в додатках на платформі .NET; XUL (XML User Interface Language) – стандарт, який розроблений в проєкті Mozilla і використовується у поштовому клієнті Mozilla Thunderbird чи браузері Mozilla Firefox; Flex – мультимедійна технологія на основі XML, розроблена Macromedia/Adobe [9].

Отже, основна ідея клієнт-серверної архітектури в тому, що існує поділ мережевих додатків, кожен з яких реалізує свій набір сервісів. Вони можуть виконуватись на будь-яких комп’ютерах, виконуючи серверні чи клієнтські функції. Така архітектура дає змогу підвищити надійність, безпеку та продуктивність мережевих додатків та мережі загалом.

Що таке “хмарні обчислення”?

Наступним еволюційним кроком у розвитку архітектурних рішень ІС є “хмари”. Використання хмарних технологій спростило організацію інформаційних процесів під час застосування великомасштабних розподілених систем. У системі хмарних обчислень динамічно розподіляються обчислювальні ресурси. Вони забезпечують ефективну координаційну роботу для виконання запитів користувачів. “Хмари” можна вважати економічною моделлю, адже вони ефективно використовують обчислювальні ресурси. Оптимізація у хмарі можлива завдяки централізованому адмініструванню, в процесі якого залучається велика кількість адміністраторів БД, а кількість мережевих і системних адміністраторів при цьому зменшується.

Відомо, що хмарні обчислення – це програмно-апаратне забезпечення сервісами, яке дає змогу клієнтам користуватись ресурсами за допомогою web-інтерфейсу чи віддаленого доступу. При цьому терміналом стає підключена до мережі робоча станція користувача, а сервери – обчислювальною хмарою. Є три сценарії консолідації: p2p (peer-to-peer) – фізичні сервери на інші фізичні сервери; p2v (physical-to-virtual) – фізичні сервери перетворюються на віртуальні; p2e (physical-to- exadata) – фізичні сервери баз даних консолідується на машини баз даних Exadata [11].

Серед перелічених найпростішими є пірингові мережі (p2p) Джона Маккарті. Приклад архітектури p2p хмарної системи наведено на рис. 7.

Відкриті хмарні системи, як і інші ІС, мають низку недоліків та переваг. Недоліками вважають: залежність користувача від провайдера хмарних послуг у питанні збереження даних. Перевагами – зниження витрат та вимог до апаратних та програмних засобів клієнтів, безпеку даних, гнучку масштабованість, економію дискового простору на ПК, зниження витрат на експлуатацію та обслуговування системи.

Раніше ми згадували про існування трьох сервісних моделей “хмари”: SaaS, IaaS та PaaS. До цього переліку слід долучити спеціалізовану версію PaaS. В Oracle виділяють модель DBaaS (Database as a Service – СУБД як сервіс). Вона містить дані замовника, які зберігаються у створеній спеціально для нього БД. Oracle пропонує два варіанти DbaaS: “чистий” – в хмарній системі замовнику створюється віртуальна машина зі встановленою СУБД; та “база даних в хмарі”. Відповідно для кожної моделі існують чотири типи реалізації хмари [7, 8]:

1. Публічні (Public Cloud) – програмні та апаратні засоби належать провайдеру ІТ.
2. Приватні (Private Cloud) – ІТ належать або взяті в оренду організацією, яка ними користується.
3. Суспільні (Community Cloud) – інфраструктура включно зі системою безпеки створена для декількох організацій.
4. Гібридні (Hybrid cloud) – комбінація з двох чи більше хмарних сервісів, пов’язаних між собою управлінськими технологіями, що забезпечує транспортування даних і додатків.

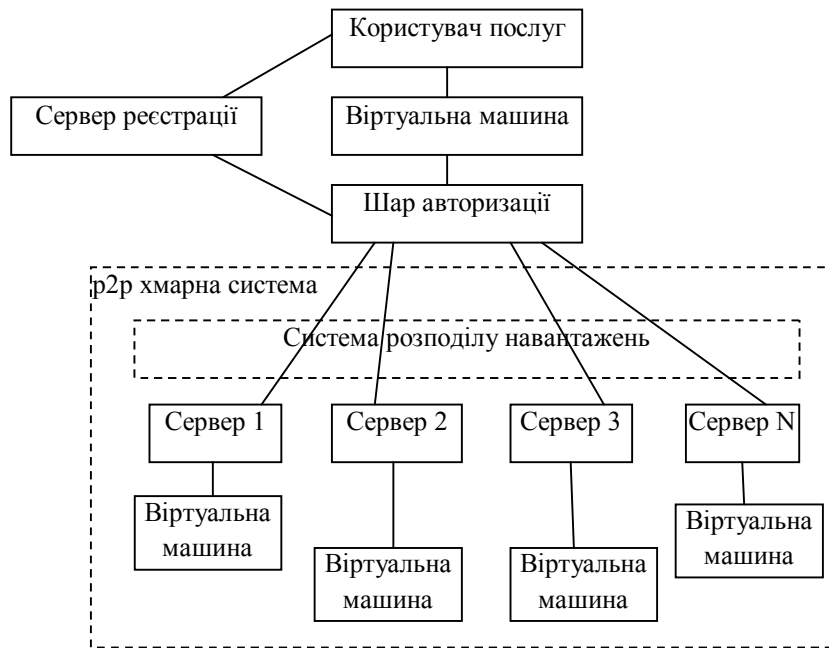


Рис. 7. Архітектура відкритої p2p хмарної системи

Сьогодні для конкурентоспроможної організації надійною щодо захисту даних є приватна хмара, яка працює на базі корпоративного центру обчислення даних (ЦОД).

Модель хмарної системи

Першочерговим при аналізі будь-якої інформаційної системи є побудова моделі цієї системи з врахуванням усіх рівнів деталізації. На рис. 8 наведено модель хмари, запропоновану National Institute of Standards and Technology (NIST) [17]. У ній координаційно виділено п'ять основних ролей: користувач хмари, провайдер хмарних послуг, провайдер доступу до хмарних послуг, аудитор хмари, хмарний брокер [15].

Користувач хмари – особа або організація, яка використовує сервіси постачальника хмарних послуг. Ця взаємодія відбувається через брокера. Залежно від потреб користувача послуг постачальник пропонує доступ до додатків у хмарі Software as a Service (SaaS), доступ до операційної системи та розробки програмних додатків Platform as a Service (PaaS) та використання віртуальних комп'ютерів та компонентів мережі Infrastructure as a Service (IaaS). Виміряти використані користувачем ресурси можна в часі, який був затрачений під час роботи додатків чи операційної системи або як загальну потужність процесора за деякий період.

Провайдер хмарних послуг – особа або організація, яка відповідає за створення та управління “хмарою” та її сервісами. Провайдер має право доповнювати хмарні послуги, змінювати їх конфігурацію, підтримувати та оновлювати роботу програмних додатків хмарної інфраструктури (SaaS). Також провайдер керує обчислювальною інфраструктурою для хмарної платформи. Він контролює програмне забезпечення компонентів “хмари” (PaaS). До його обов'язків також входить контроль за програмним забезпеченням (IaaS).

Аудитор хмари – особа або організація, яка виконує незалежну експертизу хмарного сервісу на основі стандартів відповідності побудови хмари; оцінює послуги, які надаються провайдером хмари на предмет недоторканності, продуктивності та контролю безпеки інформації користувача хмари.

Також послугою “аудит безпеки” передбачає перевірку дотримання політики безпеки та регульованих документів, перевірку відповідності чинним законам про конфіденційність, цілісність та доступність інформації на всіх етапах розроблення та експлуатації хмари.

Хмарний брокер – особа або організація, що управляє процесом використання хмарних послуг, забезпечуючи продуктивність та доставку їх до користувача і є посередником між користувачем та провайдером. Його основним завданням є полегшення взаємодії між учасниками

інформаційного процесу. Але основною функцією брокера є агрегація послуг для забезпечення інтеграції даних в ІС та їх безпечно переміщення каналами зв'язку.

Провайдер доступу до хмарних послуг – посередник, який забезпечує підключення і транспортування хмарних послуг до користувачів хмари. Вони виконують функції з надання безперервного, надійного та безпечного каналу доступу до “хмарних ресурсів”.

Аналіз моделі безпеки “хмари”

Проаналізувавши роботу основних [21, 22] учасників хмарної архітектури (рис. 8), автор пропонує розглянути формальну модель безпеки хмари, яку запропонувала NIST (рис. 9). На рисунку зображено модель хмарної системи, в окремих функціях чи областях діяльності якої розміщено компоненти безпеки [17].

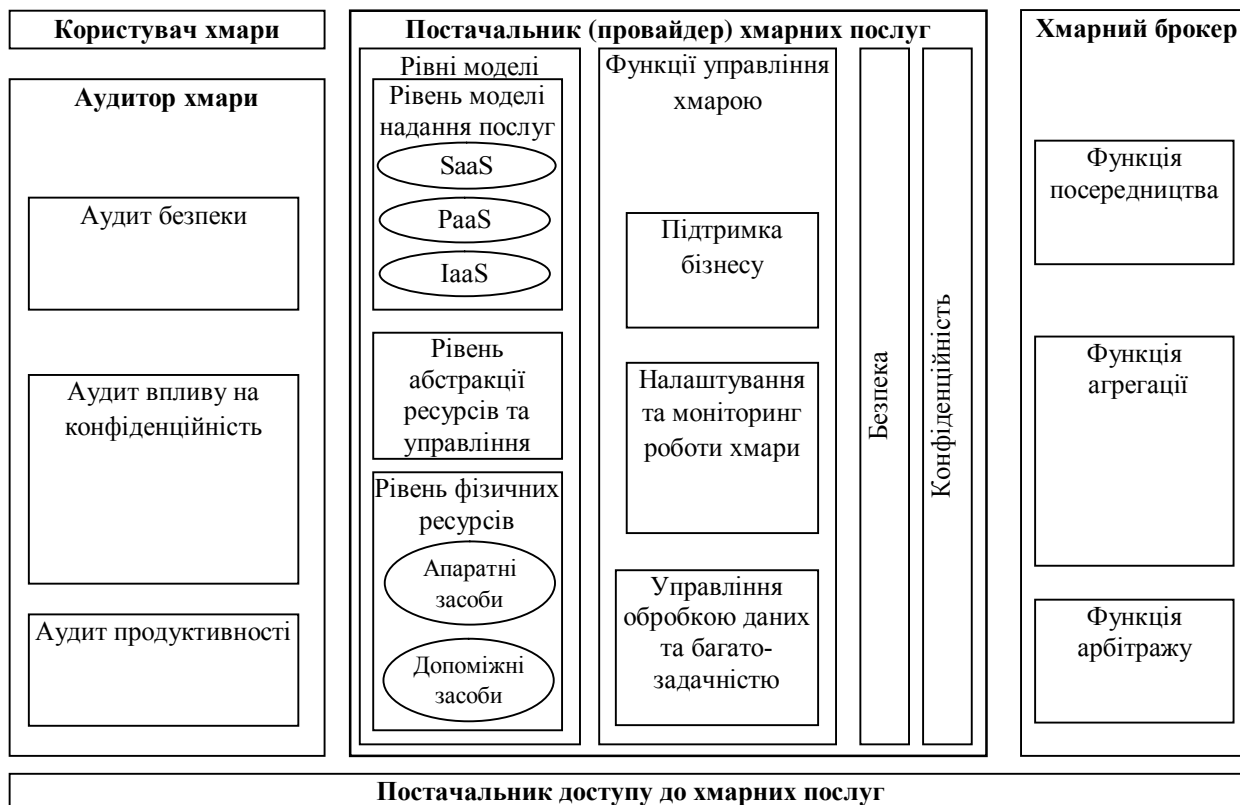


Рис. 8. Модель хмарної системи за даними NIST [17]

Управління безпекою хмари містить усі функції, необхідні для управління та роботи цього сервісу. Цей сервіс поділяється на такі компоненти [17]:

- конфігурація безпеки відповідає за дотримання стандартів безпеки, технічних умов та правил, з використанням її інструментів та політики. Цей компонент розглядає питання безпеки оперативного розгортання хмарних систем, зміни на оновлення системи. Також у ньому проводиться моніторинг, звітність, вимірювання та облік спожитих ресурсів;
- безпека функції оброблення даних та багатозадачності гарантує переміщення та розгортання між різними провайдерами хмарних послуг або брокерами даних, додатків, сервісів користувачів з дотриманням конфіденційності, цілісності та надійності;
- безпека підтримки бізнесу передбачає питання безпеки ведення ділових відносин;
- безпека організаційного забезпечення охоплює політику, процедури і процеси, що надаються провайдером для підтримки безпечного споживання хмарних послуг.

Об'єднання елементів безпеки системи, наведених на рис. 9, надають користувачам хмарні послуги, проводять координацію та управління обчислювальними ресурсами хмарних сервісів та

відповідно надають безпечні послуги користувачу. Ця інтеграція дає змогу організувати потрібну функціональність системи через інтерфейс, який надається провайдером або брокером.

Компоненти безпеки провайдера хмарних послуг визначає та контролює сам провайдер залежно від сервісів, які розміщені на всіх рівнях “хмари”. Існує два типи провайдерів: основний та провайдер-посередник. Основний провайдер надає послуги користувачам за допомогою технічного брокера або провайдера-посередника. Провайдер-посередник, окрім функцій основного провайдера, співпрацює з декількома основними провайдерами. Компоненти безпеки включаються для роботи з основним провайдером та провайдером-посередником.

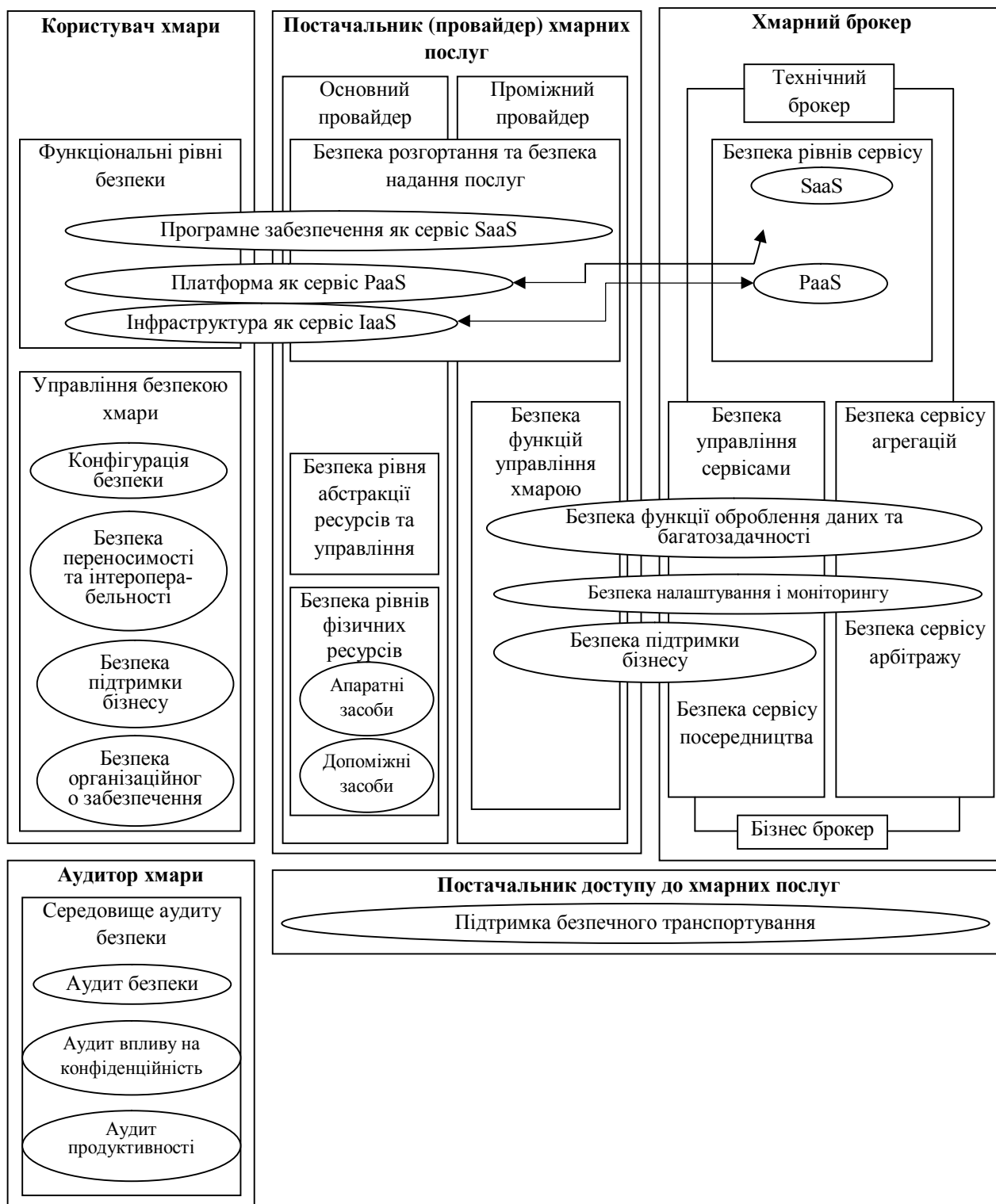


Рис. 9. Приклад формальної моделі безпеки в “хмарі” за даними NIST [17]

Відповідно до складу хмарного брокера (рис. 9) можна виділити п'ять основних компонентів безпеки [17–20]:

1) *безпека сервісу агрегації* – архітектурний компонент, що включає інтеграційні компоненти безпеки ізольованих послуг у загальносистемних послугах. Брокер забезпечує інтеграцію даних та безпечно перенесення даних від користувача до провайдера хмарних послуг;

2) *безпека сервісу арбітражу* – архітектурний компонент, який агрегує не фіксовані послуги;

3) *безпека сервісу посередництва* – архітектурний компонент, що використовує політику безпеки користувача;

4) *безпека управління сервісами* – архітектурний компонент, який включає компоненти безпеки всіх необхідних сервісних функцій для роботи брокера;

5) *безпечна екосистема хмари* – архітектурний компонент, що містить усі компоненти безпеки технічного брокера для забезпечення функціональних можливостей та реалізації додаткових послуг хмарного сервісу та у моделях розгортання хмари.

До основних компонентів безпеки для проведення аудиту належать: отримання інформації про розташування даних у системі; про результати вимірювання продуктивності споживання ресурсів; також знаходиться інформація про угоди між суб'єктами в хмарі. Вагомою складовою аудиту є інформація про механізми захисту та забезпечення конфіденційності даних у системі.

Постачальник доступу до хмарних послуг управляє компонентами підтримки безпечного транспортування, основною метою якого є забезпечення конфіденційності, цілісності та доступності сервісів, що надаються користувачеві брокером або провайдером хмарних послуг [22].

Як видно з рис. 9, безпека хмари організована на всіх рівнях сервісів. Але її можна організувати як сервіс безпеки, який можна впровадити в інформаційну інфраструктуру організації як частину сервісної шини. Також сервіс безпеки можна організувати за допомогою апаратних мережевих пристроїв з інтегрованими інтелектуальними додатками. Якщо сервіс безпеки окремо організований, то його модель може розвиватись разом з бізнес-вимогами, без впливу будь-яких додатків. Щоб розробникам сервісів розвантажитись від питань захисту в хмарі та зосередитись на роботі додатків, краще виділити безпеку як окремих сервіс.

Приклади експлуатації хмар у бізнесі

Сьогодні ринок пропонує велику кількість хмарних сервісів та продуктів. Наприклад, IaaS від Amazon Cloud EC2 і VMware або DBaaS від Microsoft Azure і Oracle DBaaS, SaaS від Salesforce і GoogleApps, PaaS розглядають з GoogleAppEngine (Java і Python) і EngineYard (Ruby on Rail). Виділені програмні рішення є не до кінця реалізованими. Щодо українських компаній, то вони пропонують переважно хмарні SaaS сервіси та лише деякі їх можливості [22].

За визначенням NIST [17], хмари повинні володіти такими властивостями: використання ресурсів на вимогу на основі самообслуговування; повсюдний мережевий доступ; об'єднання ресурсів в пул та їх багаторазова оренда; швидке, еластичне переналаштування; вимірність послуги [17]. Наведені властивості відображають п'ять груп технологій, які вже давно працюють на ринку, а їхнє спільне використання в ІС отримало назву “хмарні обчислення”.

Наприклад, організація використовує додатки з web-інтерфейсом, які встановлять на ЦОД провайдера. Вони дають їм право на адміністрування ІС. Відомо, що це є варіантом хостингу, але не хмари. Також слід розрізняти віртуалізацію та хмару. У хмарі можна використовувати віртуальні машини, але вона може існувати і без них – в рішеннях Oracle DbaaS та деяких рішеннях від Google.

Сьогодні вітчизняні ІТ-компанії пропонують такі напрямки розвитку своєї діяльності: динамічні ЦОД та приватні хмари; управління послугами; інформаційна безпека; інтеграція даних. Але така розпорошеність на ІТ-ринку не призводить до якісної роботи провайдерів. Щоб розвиватись швидше за ринок, слід визначитись з типом бізнес-моделі: продукт-орієнтована, клієнт-орієнтована, процес-орієнтована. Найкращим рішенням є продукт-орієнтована модель [14].

Для ведення ефективного бізнесу хмарні технології є послугою, методика і ІТ ресурси якої повинні взаємодіяти. Проте існують перешкоди до застосування хмари – це постійний мережевий доступ до рівня Service Level Agreement (SLA) – договір про рівень надання послуг, що є

необхідним для бізнесу замовника. Для користувачів послуг цей сервіс дозволяє моніторити формування та розподілення пулів ресурсів між користувачами. Для бізнес-моделей розподілених обчислювальних систем було сформульовано і доведено теорему CAP. Вона доводить, що користувач може реалізувати виключно дві властивості системи з трьох: Consistency (повна узгодженість), Availability (постійна доступність), Partition-tolerance (стійкість до відмови будь-яких компонент) [12].

Відомо, що у розподілених системах стійкості досягають завдяки резервуванню вузлів. В такому варіанті існує небезпека втрати зв'язку, але це не може бути причиною несправності роботи всієї ІС. Тобто можна іншими методами компенсувати доступ до додатків.

Для бізнесу основним у “хмарі” є не обчислювальні технології, а технології зберігання даних, які забезпечують швидкий, гнучкий, постійний доступ до них. Обробка даних в “хмарі” технологічно обмежена. Це зумовило появу машини обробки БД – database machines, в яких ІТ-компанії більшість функцій обчислювального рівня вбудували у систему зберігання даних. Так виник термін “великі дані”, який охоплює спробу забезпечити зберігання, індексацію, швидкий доступ до неструктурованих даних розподіленої ІС. Такі ІТ-рішення змінюють вимоги бізнесу до аналітичних систем та підходів до організації СУБД, також до широкого використання нереляційних СУБД та класу NoSQL СУБД [13].

Модернізація ІТ сьогодні влаштовує бізнес. Обслуговування ІТ-систем значно спрощує його ведення. На ринок виходять продукти з наперед інтегрованими стеками рішень. Основну увагу зосереджено на реінтеграції для спрощення інфраструктури організації та централізації її бізнес-задач. Першим кроком на шляху до централізації була візуалізація, інформаційний процес якої дає змогу консолідувати дані, а не обчислювальні потужності.

За значного накопичення даних виникає потреба у динамічному перерозподілі ресурсів. Для цього слід застосовувати передінтегровані стеки, які дають змогу моніторити інформаційні потоки та управляти ними. У приватних хмарах вони є основою.

На ринку ІТ-продуктів існує два підходи до інтеграції даних у БД. До першого належить передналаштована система, яка здатна скорочувати ризики та втрати на інтеграцію. До другого – конструктор типу “склади сам”, який має ускладнену систему та технології [14].

Складні обчислювальні комплекси складаються з чотирьох компонент: обчислювальної, мережевої, зберігання даних і СУБД. У процесі модернізації та експлуатації виникають проблеми на місцях стику цих компонент. Тому ІТ-виробники в процесі конструювання та виробництва передінтегрованих систем перевіряють інтелектуальний та технологічний потенціал інтеграційних компонент.

Наш ринок насичений великою кількістю персональних пристроїв. За їх допомогою можна доступитись до власних чи корпоративних інформаційних ресурсів. Їх інтерфейс різноманітний, адже в них працюють різні ІС. Така різноманітність ускладнює користування цими пристроями, адже для користувача не важливими є наявність інтерфейсу та оновлення версії операційної системи (ОС). Йому потрібен єдиний формат для даних, які розміщені у доступному для нього інтерфейсі.

Рішенням, яке дасть змогу ефективно і комплексно вирішити більшість зі вказаних проблем, є віртуалізація робочих місць користувачів (VDI). Сьогодні вони є альтернативою звичайним ПК.

VDI (Virtual Desktop Infrastructure) – це технологія, якою можна створити віртуальну ІТ-інфраструктуру та розгорнути повноцінні робочі місця на базі серверних систем (рис. 10). Перевагами її є [23]:

1. Робоче місце у хмарі, яке дає змогу розширити географію ведення бізнесу.
2. Централізоване та комплексне управління великою кількістю робочих місць з єдиною політикою, налаштуванням інтерфейсу, програмним забезпеченням ПК тощо.
3. Можливість створення нового робочого місця за декілька хвилин, підвищення гнучкості ПК, зміна конфігурації.
4. Усі інформаційні ресурси знаходяться на професійних сервісних платформах компанії, тому вони захищені та ізольовані, тобто консолідується корпоративні дані.

5. Підтримка роботи локальної мережі LAN та віддалений доступ WAN.

6. Підтримка аудіо- та відеоконференції.

7. Можливість використання USB-накопичувачів, підтримка смарт-карт, ключів E-token.

Також підтримується робота сканерів, принтерів та інших USB-пристроїв. У цьому архітектурному рішенні (рис. 10) концепція VDI пропонує віртуалізацію Windows XP. Управління доступом до робочих станцій та налаштування безпеки здійснюють брокери робочих станцій. Архітектурна конфігурація задається сервером з'єднання (Virtual Desktop Manager, VDM). VDM зазвичай розміщується в демілітаризованій зоні (Demilitarized Zone, DMZ) і він не включений в домен Active Directory.

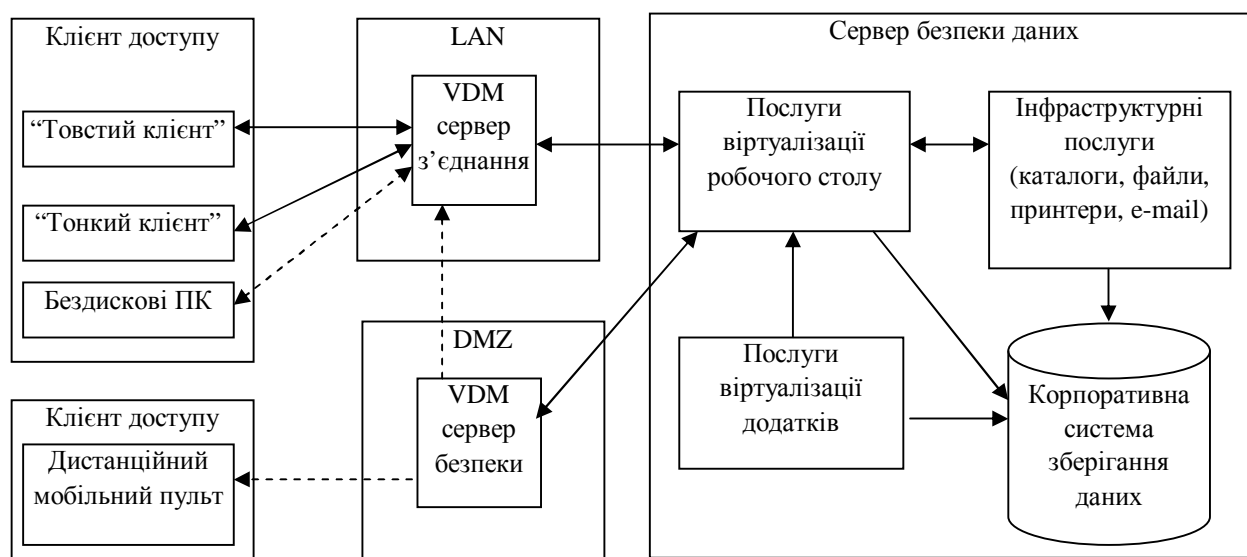


Рис. 10. Архітектура VDI технологій

DMZ – сегмент мережі, який містить загальнодоступні сервіси, наприклад web-сервіси, які допомагають відповідати на будь-які запити зі зовнішнього середовища за одночасної ізоляції локальних ресурсів (файлові сервери, робочі станції). Суть DMZ в архітектурі VDI – додатковий рівень безпеки у локальній мережі, який дає змогу мінімізувати втрати у випадку атаки на загальнодоступні сервіси, адже спочатку здійснюється прямий доступ до інформаційного та програмного наповнення DMZ [20].

VDM поділяють всіх користувачів на два типи: адміністратори та користувачі. Одні мають можливість адмініструвати свої робочі станції, інші отримують доступ до цих станцій через адміністраторів. Система VDM працює як посередник або брокер. Забезпечити доступ до сервера VDM через Internet можна за умови налаштування URL.

За дії нових парадигм у бізнесі IT поділяються на два класи задач: інженерні, які пов'язані з побудовою та експлуатацією інфраструктури та прикладні: обробка, аналіз даних та управління процесами і послугами. Сьогодні IT-ринок поділяється на Operation Support System (OSS) та Business Support System (BSS) сектори, тобто ринок виробляє IT-рішення підтримки операцій та IT-рішення підтримки бізнесу [15].

IT-ринок можна подати у вигляді двох частин: “нижня” – інфраструктурна або класична система, ціллю якої є проекти замовників для підвищення внутрішньої ефективності підприємств; “верхня” – проекти, які спрямовані на підвищення ефективності бізнесу замовника. У “верхньому” рівні можна виділити три класи рішень: згори заходяться системи обліку та Business intelligence (BI) рішення, під ними розташовуються системи класу Customer relationship management (CRM), а на нижчому рівні розташовані системи класу Business Process Management / Enterprise Content Management (BPM/ECM). У цій інфраструктурі не виділяють системи планування ресурсів

підприємства (Enterprise Resource Planning, ERP), які інтегрують бізнес-задачі, для використання в процесі інформаційної діяльності [18].

“Верхня” частина інтегрує систему оперативного управління. Вона працює з функціями та задачами в середині бізнес-процесів та зі зовнішніми інформаційними ресурсами. На “нижньому” шаблі розміщуються системи оперативного управління процесами. Вони виконують поточні завдання та оперативно обліковують інформаційні ресурси. CRM система інтегрує в собі набір додатків, функцій та інструментів, які пов’язані в єдину бізнес-логіку. Вони об’єднуються в корпоративну інформаційну систему організації. За допомогою технології ВІ обробляється великий обсяг неструктурованих даних для знаходження стратегічних можливостей бізнесу [15].

На стику “верхньої” та “нижньої” частин ІТ-ринку знаходяться проекти (Information Technology Service Management, ITSM) – це сервісно-процесний підхід до організації роботи ІТ-служби. Суть його в тому, що діяльність ІТ-підрозділу розглядається у розрізі послуг, які надаються йому на рівні домовленості між організаціями.

Також на “межі” розміщений кордон операційної та бізнес-діяльності організації, наприклад, це проекти уніфікованої комунікації та відеоконференцзв’язок. Тому слід триматись ближче до “межі”, розширюючи компетенцію процесного консалтингу через ITSM, який присутній в BPM / ECM та CRM управлінських рішеннях [21].

При впровадженні інфраструктурних рішень основним фактором є вендори, які контролюють ринок системної інтеграції. Адже вендор (vendor) – це компанія-постачальник товарів та послуг під своєю торговою маркою. В них зосереджена бізнес-модель, в якій встановлюється зв’язок користувача від товарів чи послуг.

Зараз на ІТ-ринку спостерігається дозрілий попит. Провайдери ІТ-послуг чітко позиціонують свої послуги відповідно до потреб бізнес-структур. Сьогодні ІТ-проблеми стають проблемами бізнесу. Ринок системної інтеграції досяг рівня зрілості. Деякі гравці ІТ-ринку оптимізують свою структуру під системні інтегратори, а інші шукають можливості для об’єднання інфраструктурних та бізнес-систем.

На українському ринку ІТ-продуктів над “межею” опинився процесний консалтинг у вигляді сервісів, які надають ITSM, BPM або CRM. Бізнес-системи починають “мислити” процесами і сервісами. Адже ІТ-сервіси можна отримати з хмари: аудіо-, відеозв’язок, пошта, CRM, сервісдеск, файлове зберігання. Потрібно лише домовитись про рівень компенсації та політику безпеки [19, 20].

Сьогодні найперспективнішими є публічні хмари, і попит на них зростає. Приватні хмари дозволяють здійснити перехід бізнес-моделі від розподіленої ІТ-системи до централізованої. Головним питанням переходу до приватної хмари є етапність цього процесу, адже в цьому сегменті спостерігається вузький вибір ІТ-рішень. Кращим рішенням є створення інфраструктури підприємства з нуля. Адже коли інфраструктура існує, то вона організована на підприємстві централізовано, інформаційні ресурси в ній віртуалізовані, а ІТ-сервіси – статичні. Виникає питання: чи потрібно для даного підприємства організувати приватну хмару?

Сьогодні основним напрямом діяльності ІТ-провайдерів є побудова інфраструктур приватної хмари IaaS на рівні підприємств. В них використовується архітектура OpenStack, яка є стандартом де-факто для побудови хмарних систем зберігання та обробки даних [19].

OpenStack – це платформа з відкритим вихідним кодом для хмарних обчислень, які підтримуються корпорацією IBM® та іншими галузевими лідерами. Вона пропонує операційну платформу для широкомасштабних хмарних рішень. Також платформа використовує гіпервізори та програмні засоби для віртуалізації робочих машин на стандартних апаратних засобах.

OpenStack платформи складаються з семи модулів: Compute (Nova); Networking (Neutron/Quantum); Identity Management (Keystone); Object Storage (Swift); Block Storage (Cinder); Image Service (Glance); User Interface Dashboard (Horizon) [14].

Модуль OpenStack Compute (Nova) є базовим компонентом інфраструктурних сервісів, який написаний мовою Python. Він створює рівень абстрагування для віртуалізації ресурсів серверів (процесори, пам’ять, мережеві адаптери, жорсткі диски тощо). Модуль підтримує функції для підвищення коефіцієнта використання та адаптації системи [17].

Модуль Networking (Neutron) забезпечує управління локальними мережами, які підтримують віртуальні системи (VLAN), протоколи DHCP та IP v6. У цьому модулі користувач може визначати мережі, підмережі, маршрутизатори, призначати їм IP-адреси та створювати власні віртуальні мережі.

Модуль OpenStack Identity Management (Keystone) керує каталогом користувачів, а також каталогом сервісів OpenStack. Його метою є за допомогою плагінів підтримувати централізовану аутентифікацію (використання логінів/паролів або багатофакторних систем) [12].

Модуль OpenStack Object Storage (Swift) являє собою резервну систему зберігання даних для горизонтального масштабування зберігання ресурсів. Модуль гарантує реплікацію даних та їх розподіл в пулі між пристроями.

Модуль OpenStack Block Storage (Cinder) керує сховищем блочного рівня, який застосовує екземпляри Compute.

Модуль OpenStack Image Service (Glance) забезпечує підтримку образів віртуальних машин, подекуди системних дисків, які призначені для запуску віртуальних машин.

На рис. 11 зображено архітектуру OpenStack платформи, в якій три її елементи взаємодіють зі всіма компонентами системи. Це Horizon, який є графічним інтерфейсом користувача та дає адміністраторам змогу керувати всіма проектами. Другим елементом є Keystone, що управляє авторизованими користувачами. Третій – Neutron – визначає мережі, що забезпечують зв'язки між компонентами системи.

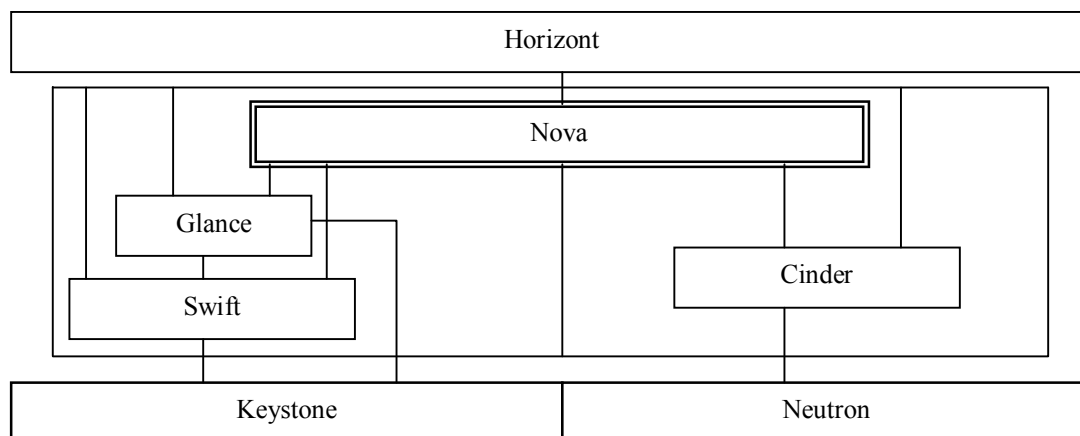


Рис. 11. Архітектура OpenStack платформи

До центрального елемента архітектури OpenStack, який наведено на рис. 11, належить Nova. До її функцій належить розпаралелення робочих навантажень. Елементам Compute потрібно сховище даних, яке може бути або хмарним Cinder, або об'єктним Swift. Окрім сховища, елемент Nova потребує образу для запуску екземпляра. Відповідно Glance обробляє цей запит. При цьому як внутрішнє сховище використовують Swift [19].

Архітектура OpenStack підтримує незалежність кожного проекту, що, своєю чергою, дає змогу розробляти лише потрібну підмножину функціональності та інтегрувати її до інших систем чи технологій, які пропонують подібні функції. Лише повне об'єднання всіх функціональних елементів архітектури OpenStack гарантує повнофункціональну діяльність приватної хмари.

Сьогодні на українському IT-ринку зосереджено увагу на технології класу Big Data для управління накопиченими даними підприємства великого обсягу. Означення Big Data містить визначення трьох технологій: Volume – обсяг даних, Velocity – швидкість обробки інформації, Variety – багатообразні та неструктуровані дані. Для цього технологічного рішення слід застосувати архітектуру Hadoop, організована робота якої дає можливість бізнес-структурам обробляти великі масиви інформаційних потоків, виділяючи при цьому важливі дані. Організаційна схема архітектури допомагає вибудовувати стратегію підприємства, нарощувати та управляти своєю клієнтською базою, підвищувати його рентабельність [17].

Технологія Hadoop являє собою програмну платформу (framework), яка за допомогою комп'ютерних кластерів та парадигми MapReduce дає змогу зберігати та обробляти дані підприємства. Такий підхід забезпечує побудову високовиробничих кластерів на базі low-end або middle-end серверів, які впливають на ціну IT-ресурсу.

В основу технології покладено розподілену файловою систему HDFS (Hadoop Distributed File System), що забезпечує зберігання даних на деяких вузлах кластера. Тому робота БД Hadoop після збою одного чи декількох вузлів мінімізує втрату інформації, і робота системи продовжується в штатному режимі [3].

Важливим елементом Hadoop є використання в її основі парадигми MapReduce, яка є програмною платформою для обчислень у розподіленій мережі. Обробка інформації у цій платформі здійснюється в два кроки: Map – здійснення первинної обробки даних на всіх вузлах та Reduce – зведення обробленої інформації для отримання результату. В табл. 1 наведено найважливіші елементи системи Hadoop.

Технологія Hadoop застосовується для аналізу великих обсягів корпоративних даних, також інформації отриманої з додаткових джерел. Її використання дає змогу скоротити час на оброблення і підготовлення даних для аналітичної системи. Також технологія розширює можливості для аналізу слабоструктурованих або неструктурованих даних. Вона дає змогу оптимізувати інформаційні процеси та скоротити витрати на зберігання й обробку даних, забезпечуючи ефективність роботи з даними.

Таблиця 1

Опис утиліт Hadoop системи

Назва утиліти	Опис властивостей
HBase	NoSQL БД для зберігання великого обсягу даних
Hive	Настройка для Hadoop, в якій дані зберігаються у вигляді таблиць, а звернення до них можливе через синтаксис SQL
Pig	Настройка для Hadoop, що дозволяє створювати програми для зберігання даних високорівневою мовою
Mahout	Набір бібліотек, які використовують в MapReduce та застосовують алгоритми Data Mining
Rhadoop	Настройка над Hadoop для виконання програм, написаних мовою R. Дозволяє використовувати алгоритми Data Mining
Oozie	Утиліта створює workflow з програми MapReduce та запитів Hive и Pig
Hue	Web-інтерфейс до Hadoop для моніторингу завдань
Sqoop	Утиліта для завантаження даних з реляційної БД в кластер Hadoop та для відправлення їх назад у БД

Можна використати технологію Hadoop для підготовки даних в аналітичній платформі, з проведенням обчислень в OLAP-кубі та з використанням Data Mining. На рис. 12 наведено приклад такого виду архітектури [4].

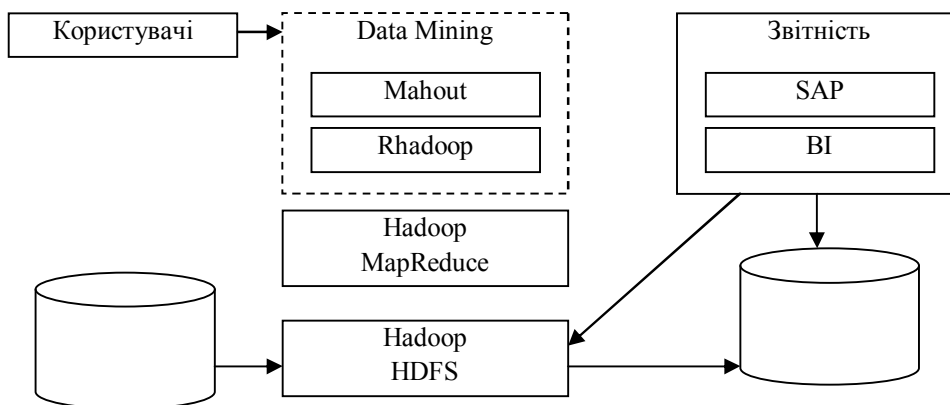


Рис. 12. Архітектура аналітичної платформи Hadoop

Це архітектурне рішення (рис. 12) застосовують для обчислень віртуальних даних з метою створення звітів. Використання аналітичної платформи є перспективним напрямом розвитку інтелектуального аналізу даних у хмарі, застосування якої підвищує відмовостійкість та легке масштабування ІС. Інтегрування приведених в табл. 1 утилітів оптимізує роботу платформи, де не останню роль відіграє використання технології Ві-звітності.

Об'єднання існуючої інфраструктури без доопрацювань з боку Ві-системи з використанням попередньої архітектури наведено на рис. 13.

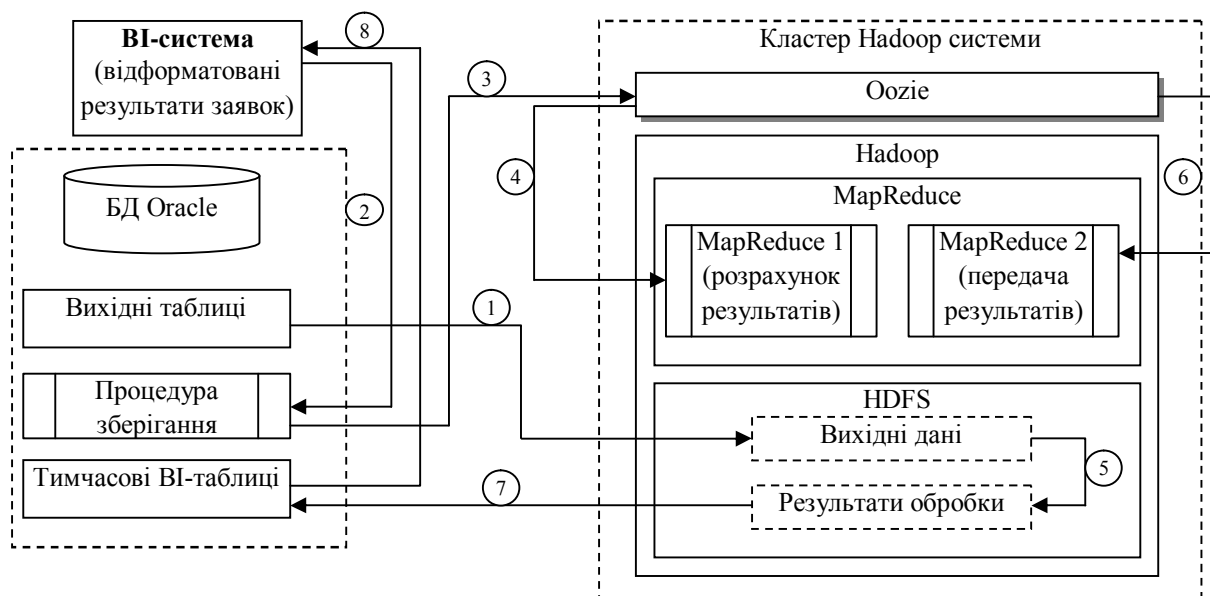


Рис. 13. Архітектура технології Hadoop

На рис. 13 інформаційний процес за цією технологією організовано так: спочатку, на першому кроці завантажуються дані в HDFS файловою системою, яка призначена для зберігання файлів великих розмірів, що розподіляються між вузлами кластера обчислень.

На другому кроці викликається процедура Ві-системи, яка працює у БД з відформатованими результатами. На третьому кроці з процедури зберігання БД викликається потік робіт до утиліти Oozie. Четвертий крок – виклик процедур від утиліти Oozie до MapReduce1. П'ятий – в HDFS з блоку Вихідні дані надходить інформація в результат обробки. Шостий – від блоку Oozie надходить потік робіт у MapReduce2. На сьомому кроці переміщуються результати обробки до тимчасових таблиць БД. На восьмому кроці дані в тимчасових таблицях форматуються та надходять до інтелектуальної Ві-системи [7].

На рис. 13 реалізовано інтерфейс завантаження даних з БД Oracle в кластер Hadoop. Така співпраця є перспективною, адже проводиться якісний процес обчислення інформації, який для подальшої обробки надходить в Ві-систему. Таке архітектурне рішення технології Hadoop дозволяє зберегти всі переваги від використання Ві-системи.

До основних перспективних рішень, які було прийнято в процесі використання кластера обчислень, належать такі: економія часу на обробку даних; відносно низька вартість технічних засобів; підвищує виробництво системи; лінійне масштабування дозволяє збільшувати кількість вузлів у кластері; ця технологія дозволяє здійснювати складну обробку файлів структурованої чи неструктурованої структури, тому дані можуть ефективно оброблятися та використовуватися в системі.

Інтерес до OpenStack і Hadoop є не випадковим, адже ці стандарти впорядковують підходи до побудови архітектур. З їх використанням можна нарощувати компетенцію в інтеграції компонент цих архітектур між собою та з компонентами ІТ-систем інших підприємств [4].

Платформа Hadoop є відкритою, але щоб з нею працювати та підключати різні джерела даних, потрібні фахівці. На сучасному українському IT-ринку їх не так і багато. Тому бізнес-структури спростили задачу і у більшості випадків зорієнтувались на платформу Teradata [2].

Ця платформа містить у своєму арсеналі інструменти, які оперативно встановлюють зв'язок з новими джерелами даних, які працюють із звичною мовою SQL. Архітектуру платформи наведено на рис. 14.



Рис. 14. Архітектура платформи Teradata

Архітектура Teradata містить потужні СУБД, які підтримують великі БД (Very Large Databases – VLDB), з якими працюють різноманітні додатки з підтримки та прийняття рішень (Decision Support System – DSS) і додатки сховищ даних (Data Warehousing – DW). Архітектура платформи побудована за принципом “нічого не розділяти”. Структура СУБД платформи організована за принципом паралельного передавання багатовимірних даних. Робота архітектури не підтримує однопотокового передавання даних. Проектуючи додатки та утиліти, беруть до уваги паралелізм інформаційних процесів [2].

У Teradata паралельно виконуються всі SQL-оператори: форматування, відновлення, завантаження та видалення даних, управління пріоритетами, інсталяція та поновлення програмного забезпечення, відлагодження, доступ до словників, блокування з'єднання таблиць та інші задачі.

СУБД Teradata володіє образом БД та використовує деякі віртуальні процеси (Vprocs). Взаємодіючи, вони використовують шар високошвидкісного передавання даних (Message Passing Layer) з незначним часом затримки. Цей шар використовують в складі операційної системи для паралельних БД (Parallel Database Extensions, PDE). При передаванні даних через PDE підтримуються два типи віртуальних процесів: PE (Virtual Parsing Engines – Virtual PE), які відповідають за взаємодію та управління користувачькими сесіями, оптимізацію запитів, створення та управління планом запитів; AMP (Virtual Access Module Processes – Virtual AMP), які використовуються для фактичного маніпулювання таблицями БД. Отже, Teradata – це паралельна реляційна СУБД, яка працює на операційних системах [16].

Великі корпорації все частіше зосереджують свою увагу на системи побудови інфраструктури приватної хмари та на технології “великих даних”. Тому вони переходять від стандартів ITSM/ITAM та BPM, плануючи користуватись BSC, які інтегрують нові рішення з наявними системами. Сервісна і процесна орієнтованість ITSM знаходить своє застосування в бізнес-середовищі.

У деяких традиційних фінансових інструментах вже реалізовані модулі ITSM та ITAM, зближуючи бізнес-процесний та IT-сервісні види управління. Такий рух призводить до управлінського підходу BPM, оскільки у нього введено управлінські методології.

Використання BPM проектів підвищується з рівнем автоматизації та одночасним збільшенням кількості програних продуктів, які взаємодіють між собою. Позитивним є те, що при інтеграції додатків з використанням BPM-систем є очевидна направленість інтеграції на вирішення бізнес-проблем замовника.

Сьогодні небагато компаній впроваджують у свою діяльність складні проекти з ITSM/ITAM. На ці ключові концепції попит зростає, він трансформується у широку компетенцію управління послугами на базі методології процесного підходу [9].

Спільне у ITSM і хмар є те, що вони орієнтуються на сервіси, адже хмарні обчислення будь-які IT-ресурси представляють у вигляді послуги. Тому використання методології ITSM є невід’ємною частиною технології та методології надання хмарних послуг. Адже виробники систем управління хмарною інфраструктурою включають у свій комплекс ITSM компоненти.

Хмари від Oracle

Аналізуючи вищенаведений матеріал, можна стверджувати, що найкращим рішенням для сучасного бізнесу є використання хмарних обчислень. Приватна хмара дозволяє зосереджувати сучасний інструментарій та методику для вирішення управлінських завдань. Сервіси приватних хмар підтримують моделі IaaS, PaaS, SaaS і DBaaS. Тому замовнику надається технічний набір продуктів та технологій для підтримки життєвого циклу інформаційних потоків [9].

Хмари від Oracle дозволяють компаніям розгортати не тільки окремі віртуальні машини або БД, а й мультимедійні комплекси. Найчастіше використовуються моделі DBaaS від Oracle, які представлені трьома реалізаційними варіантами, наведеними на рис. 15.

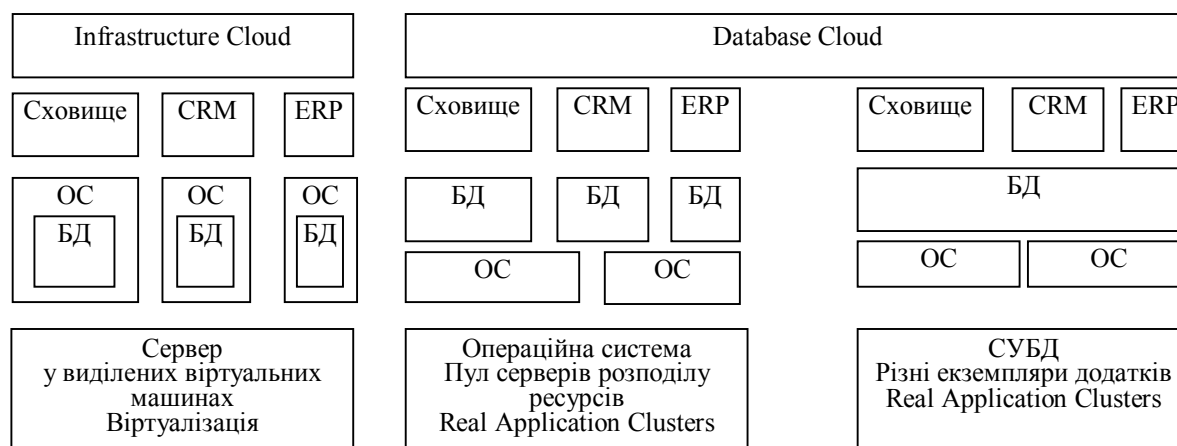


Рис. 15. Три варіанти реалізації dBaaS в хмарі Oracle

До першого варіанта належать: Database in VM або Infrastructure Cloud, які працюють на окремих віртуальних хмарах. Процес перенесення БД в хмару Oracle є нескладним, адже кожна віртуальна машина має власну ОС та БД і свою версію СУБД. Усі машини розміщуються в одному серверному пулі, тим самим підвищуючи ефективність використання ПЗ. А використання СУБД за цим варіантом є неефективним, бо системі потрібно підтримувати багато ОС та БД. Вона підходить для першого кроку консолідації системи [6].

Другий варіант називається “чистий” DBaaS, в якому паралельно працюють уніфікована ОС та СУБД Oracle. Така архітектурна інфраструктура спрощує експлуатацію ПЗ системи. В ній екземпляри кластера СУБД працюють на пулі фізичних ПК з єдиною ОС [6].

За третім варіантом інтегруються різні додатки в єдину кластерну базу. Організована хмарна БД забезпечує додаткові засоби розмежування доступу та зберігання інформації. Прикладом такої архітектурної консолідації є приватна хмара Oracle [6].

Приватна хмара в Oracle складається з таких етапів життєвого циклу: планування структури хмари та варіанти її консолідації; створення хмарної інфраструктури; підготовка ПЗ; тестування

сервісів та їх публікація; замовлення та використання сервісів, їх моніторинг; управління хмарною інфраструктурою, тарифікація і білінг використаних ресурсів; оптимізація використання ресурсів.

При роботі у приватній хмарі Google замовник через браузер на порталі самообслуговування створює заявку на технічну та програмну інфраструктуру ІС. Він не знайомий з інфраструктурою самої хмари її конфігурацією ОС та ПЗ тощо.

Для ефективної організації бізнес-моделі слід використовувати приватну хмару з моделлю побудови DBaaS на машині Oracle Exadata, адже вона може підтримувати програмне забезпечення СУБД на будь-яких машинах. За такою архітектурою встановлюються пріоритети системи введення/виведення інформаційних потоків для БД, а також зручна конфігурація системи зберігання та вузлів кластера чи мережевих елементів [16].

Висновки та перспективи подальших наукових розвідок

Проаналізовано еволюцію розвитку архітектур інформаційної системи бізнес-моделі. В процесі розвитку кожна наступна архітектурна структура вирізнялась ефективнішим використанням ресурсів підприємства та призначалась для забезпечення надійного, масштабованого та доступного зв'язку з комунікаційною мережею на фізичному та логічному рівнях. Для гарантування безпеки ПЗ мережевих служб враховувались вимоги з безпеки на структурному та логічному рівнях конфігурації. Тобто, на етапі проектування архітектури ІС брались до уваги вимоги з безпеки мережі. Також важливою при виборі архітектури ІС є система управління бізнес-моделлю.

Сьогодні з появою хмар ІТ-індустрія стоїть на порозі інноваційних змін. Така еволюція дається непросто, адже процес міграції додатків у хмару відбувається не без проблем.

Хмарний підхід дозволяє матеріалізувати дві основні ідеї: комунальної обробки інформації та самоуправління. Разом з хмарами підприємство набуває нової якості і стає підприємством наступного покоління (Next Generation Enterprise). Воно володіє такими якостями, як: динамічність, адаптивність, глобальна інтеграція, прозорість бізнес-структури. Ці якості не є новими, їх протягом всієї еволюції архітектур ІС пробували прищеплювати організаціям. Відсутність необхідної платформи тільки ускладнювала процес реалізації нових якостей. Тільки з хмарами та сервісними обчисленнями з'явилась змога створити архітектуру ефективної, динамічної, мережевої ІС.

Існує одвічне питання: як пов'язані між собою підприємства нової генерації, архітектура та хмари? Відповідь доволі проста – вони доповнюють один одного, адже хмари дозволяють перекрити простір між корпоративною архітектурою та інфраструктурою, вона здійснює інтеграцію модулів КІС через Інтернет для створення підприємств Next Generation Enterprise. Хмари консолідує всі рівні моделі корпоративної архітектури, адже немає підходів, котрі б від мейнфреймів до інтернет-технологій забезпечували комплексним вирішенням поставлених задач.

Сучасні хмарні технології є перспективним рішенням та елементом “третьої ІТ-платформи”. Тому сучасним підприємствам слід вивчати системний та комплексний розвиток хмарних сервісів.

Використання хмарних технологій пов'язане зі зменшенням витрат на інтенсифікацію. При тому хмарні рішення постійно вдосконалюються і можуть досягати високого рівня безпеки.

1. *A Practical Distributed Mutual Exclusion Protocol in Dynamic Peer-to-Peer Systems : (3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)) [Електронний ресурс] / L. Shiding, C. Ming, Z. Zheng, L. Qiao. – Режим доступу до журн. : <http://www.sciweavers.org/publications/practical-distributed-mutual-exclusion-protocol-dynamic-peer-peer-systems>.*
2. *Advanced Peer-to-Peer Networking (APPN) [Електронний ресурс]. – Режим доступу: http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp?topic=/com.ibm.zos.znetwork/znetwork_181.htm.*
3. *Allen F. BlueGene: A Vision for Protein Science Using a Petaflop Computer [Електронний ресурс] / F. Allen // IBM Systems J. – 2001. – Vol. 40, № 2. – Режим доступу: www.research.ibm.com/journal/sj/402/allen.html.*
4. *Catlett C. Metacomputing / C. Catlett, L. Smarr – Comm. ACM, 1992. – P. 128–136.*
5. *Christensen E. Web Services Description Language 1.1. – W3C Note [Електронний ресурс] / E. Christensen. – Режим доступу: <http://www.w3.org/TR/wsdl>.*
6. *Foster I. A*

Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems / I. Foster, N. Karonis // *Proceedings of SC'98*. – ACM Press, 1998. – P. 26–45. 7. Foster I. Software Infrastructure for the I-WAY High Performance Distributed Computing Experiment : (Proc. 5-th IEEE Symposium on High Performance Distributed Computing) [Электронный ресурс] / I. Foster, J. Geisler, W. Nickless, W. Smith, S. Tuecke. – 1997. – P. 562–571. – Режим доступа: <ftp://ftp.globus.org/pub/globus/papers/isoft.pdf>. 8. Foster I. The Anatomy of the Grid: Enabling Scalable Virtual Organizations : (International. J. High Performance Computing Applications) [Электронный ресурс] / I. Foster, C. Kesselman, S. Tuecke. – 2001. – 15(3). – P. 200–222. – Режим доступа: <www.globus.org/research/papers/anatomy.pdf>. 9. Foster I. The Grid: Blueprint for a New Computing Infrastructure / I. Foster, C. Kesselman. – San Francisco: Morgan Kaufmann, 1999. – 368 pp. 10. Foster I. What is the Grid? A Three Point Checklist [Электронный ресурс] / I. Foster. – Режим доступа: <http://www.globus.org/research/papers/WhatIsTheGrid.pdf>. 11. Improving P2P in Cloud Computing based on Performance by Sarada B. S. [Электронный ресурс]. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.206.3360>. 12. JMaay : p2p Java Library [Электронный ресурс]. – Режим доступа: <http://sourceforge.net/projects/jmaay/>. 13. JXTA 2: A high-performance, massively scalable P2P network [Электронный ресурс]. – Режим доступа: <http://www.ibm.com/developerworks/library/j-jxta2/>. 14. Legion: A Worldwide Virtual Computer [Электронный ресурс]. – Режим доступа: <http://legion.virginia.edu/index.html>. 15. National Institute of Standards and Technology [Электронный ресурс]. – Режим доступа: <http://www.nist.gov/>. 16. Rowstron A. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems / A. Rowstron, P. Druschel. – MIDDLEWARE, 2001. – P. 329-350. 17. Sterling T. The Gilgamesh MIND Processor-in-Memory Architecture for Petaflops-Scale Computing : (ISHPC Conference. – Kansai (Japan). – 2002, May) [Электронный ресурс] / T. Sterling. – Режим доступа: www.sc-2002.org/paperpdfs/pap.pap105.pdf. 18. The Globus Toolkit [Электронный ресурс]. – Режим доступа: <http://www-unix.globus.org/toolkit/>. 19. XMPP Standards Foundation [Электронный ресурс]. – Режим доступа: <http://xmpp.org/about-xmpp/faq/>. 20. Затуливетер Ю. С. Информация и эволюционное моделирование : (труды Междунар. конф. “Идентификация систем и задачи управления” (SICPRO'2000)) [Электронный ресурс] / Ю. С. Затуливетер. – М.: Ин-т проблем управления РАН. – 2000, 26–28 сентября. – С. 1529–1573. – Режим доступа: http://zvt.hotbox.ru/1529_.htm; http://zvt.by.ru/1529_.htm. 21. Затуливетер Ю. С. О компьютерных проблемах формирования единого информационного пространства виртуальных предприятий. Системы проектирования, технологии подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/ PDM-2002) : (материалы 2-й Междунар. конф. (CAD/CAM/PDM-2002) В 2 т. – Т.1) [Электронный ресурс] / Ю. С. Затуливетер. – М.: Ин-т проблем управления РАН. – 2002. – С. 165–176. – Режим доступа: [Z3okl_1.htm](http://zvt.by.ru/Zatuliveter_dokl_1.htm); http://zvt.by.ru/Zatuliveter_dokl_1.htm. 22. Ильин В. Российский сегмент глобальной инфраструктуры LCG / В. Ильин, В. Кореньков, А. Солдатов // *Открытые системы*. – М. : 2003. – № 1. – С. 59–68. 23. Орлов С. Искусство объединения / С. Орлов // *LAN*. – М. : 2003. – № 9. – С. 98–124.