Лекція 14.
Архітектурні моделі Big Data. Технології віртуалізації. Гіпервізори. Контейнерна технологія виконання програмного коду на сервері. SaaS, PaaS і IaaS

План лекції

14.1. Архітектурні моделі інженерії Big Data.
14.2. Центри обробки даних та хмарні обчислення.
14.3. Технології віртуалізації.
14.4. Шари абстракції.
14.5. Гіпервізори.
14.6. Контейнерна технологія виконання програмного коду на сервері.
14.7. Інжиніринг даних.

14.1. Архітектурні моделі інженерії Big Data

Перетворення даних у цінну інформацію потребує обчислювальної потужності та пам'яті. Різні архітектури IoT мають різні підходи щодо того, де і коли дані обробляються та зберігаються (рис. 14.1).

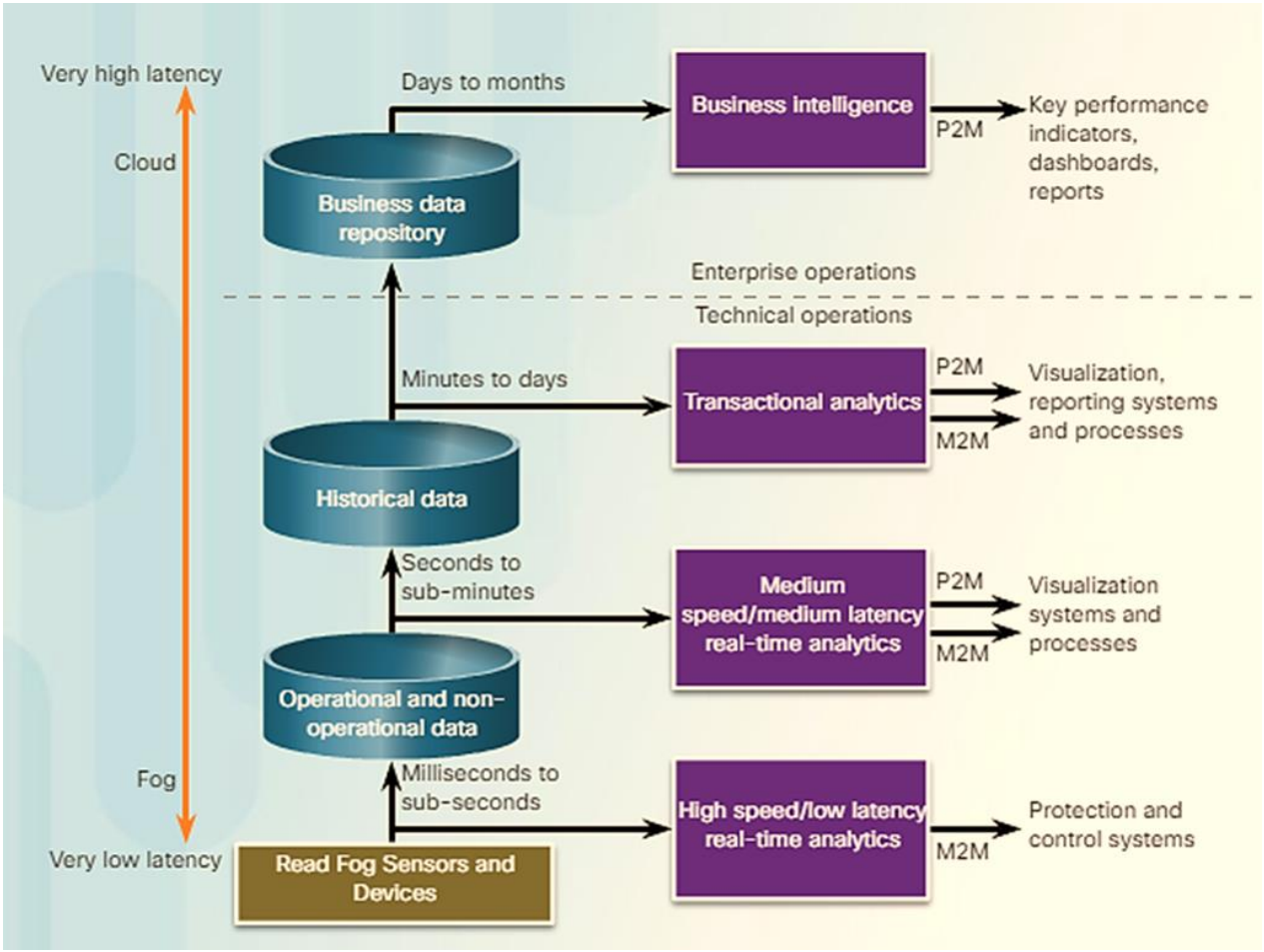

Рис. 14.1. Аархітектурна модель Device-Network-Cloud [1]

Наприклад, в архітектурній моделі Device-Network-Cloud усі точки даних, зібрані датчиками, що входять до підключеного пристрою, надсилаються безпосередньо в хмару для зберігання та оброблення.

Наприклад, дані, зібрані пристроєм для відстеження фітнес-занять, передаються до хмари. Там вони трансформуються за допомогою описової аналітики і подаються користувачеві у вебпрофілі. Ця архітектурна модель проста, але не масштабована. Коли кількість датчиків збільшується разом із кількістю генерованих точок даних або коли обробка даних вимагає значно коротшого часу відгуку, це ситуації, коли дані потрібно обробляти ближче до місця їх генерування. Тут використовується архітектура Device-Gateway-Network-Cloud. Залежно від програми дані можуть бути оброблені майже відразу після їх генерування, поблизу від джерела їх створення на шлюзі або інших проміжних місцях у мережі (туманні обчислення). Ця область джерела також відома як край, і з цієї причини такий підхід також називають крайовою аналітикою. Прикладами програм, які потребують обчислення туману, є мережі датчиків, які географічно розподілені, наприклад, датчики вологості землі на винограднику та датчики руху на кожному перехресті у місті. Туманні обчислення допомагають скоротити час відгуку (низька затримка) та зменшити кількість даних, які необхідно надсилати до хмари. Наприклад, обчислення туману видаляє надлишкові дані, коли змінна датчика не змінюється, оскільки не раціонально продовжувати передавати одні і ті ж значення. Натомість дані передаються лише тоді, коли їх значення змінюються.

Незалежно від використовуваної архітектури IoT, більшість або усі дані з часом будуть збиратися та зберігатися в хмарі, де є обчислювальна потужність та ємність для зберігання. Мережі центрів обробки даних використовують технологію віртуалізації. У центрах обробки даних доступні тисячі (або сотні тисяч) серверів, а також накопичувач потужності для зберігання даних через високошвидкісні мережеві з'єднання. Технологія віртуалізації дозволяє створювати всередині кожного фізичного сервера одну або кілька віртуальних

машин, де може працювати процес аналізу даних. Провайдери хмарних платформ мають мережу центрів передачі даних про несправності.

Дані рухаються по мережевій інфраструктурі. Організації залежать від своїх IT-операцій. Здатність інфраструктури швидко робити доступними ресурси безпосередньо впливає на швидкість передачі даних. Використання великих даних для отримання інформації та розуміння бізнесу вимагає потужних рішень, таких як центри обробки даних (ЦОД). Наприклад, по мірі розвитку організацій вони потребують збільшення кількості обчислювальної потужності та місця на жорсткому диску. Якщо залишити його без розгляду, це негативно вплине на здатність організації надавати життєво важливі послуги. Втрата життєво важливих послуг означає нижчу задоволеність клієнтів, менший дохід, а в деяких ситуаціях і втрату майна.

Великі підприємства, як правило, володіють ЦОД для управління потребами організації сховища та доступу до даних. У ЦОД один орендар підприємства є єдиним замовником, що користується послугами ЦОД. Однак, оскільки обсяг даних продовжує розширюватися, навіть великі підприємства розширюють свої можливості зберігання даних, використовуючи послуги ЦОД, які можуть використовуватися для задоволення внутрішніх потреб IT (приватна хмара). ЦОД можуть також пропонувати ці самі товари та послуги іншим компаніям та організаціям (публічна хмара).

14.2. Центри обробки даних та хмарні обчислення

Щоб допомогти вирішити чотири Vs Big Data (обсяг, різноманітність, швидкість та правдивість), багато організацій звертаються до хмарних обчислень. Хмарні обчислення підтримують різноманітні проблеми управління даними:

- доступ до даних організації будь-де та в будь-який час;
- упорядкування IT-операцій організації, підписка лише на потрібні послуги;

• зменшення потреб в IT-обладнанні, технічному обслуговуванні та управлінні на місці;

• зменшення витрат на обладнання, енергію, фізичні потреби в установках та потреби в навчанні персоналу;

• швидке реагування на збільшення потреб у обсязі даних;

• витрати на обчислення та зберігання, а не інвестування в інфраструктуру, капітальні витрати перетворюються на операційні видатки.

Три основні послуги хмарних обчислень, визначені Національним інститутом стандартів та технологій (National Institute of Standards and Technology, NIST) у спеціальній публікації 800-145:

• **SaaS** – програмне забезпечення як послуга (Software as a Service);

• **PaaS** – платформа як послуга (Platform as a Service);

• **IaaS** – інфраструктура як послуга (Infrastructure as a Service).

Провайдери хмарних послуг розширили цю модель, щоб забезпечити IT-підтримку для кожної з хмарних обчислювальних служб (ITaaS).

Наразі у світі існує понад 3000 центрів обробки даних, які пропонують загальні послуги розміщення і оброблення даних для організацій. Існує набагато більше центрів обробки даних, які належать приватній галузі та управляються ними для власного використання.

Дані центри – це централізовані місця, що містять велику кількість обчислювальної та мережевої техніки. Це обладнання використовується для збору, зберігання, обробки, розповсюдження та надання доступу до величезної кількості даних. Основна його функція – забезпечити безперервність бізнесу, зберігаючи обчислювальні послуги доступними, коли і де вони потрібні.

Практично кожна організація потребує власного ЦОД або доступу до ЦОД. Деякі організації будують і підтримують власні центри обробки даних. Інші організації орендують сервери у місцях спільного розташування. Є й інші, які використовують публічні, хмарні сервіси. Веб-сервіси Amazon, Microsoft Azure, Rackspace та Google – приклади компаній, які надають публічні хмарні послуги.

Через операційну складність ЦОД дуже мало організацій управляють власним ЦОД. Тому багато організацій орендують простір у спеціалізованих центрах даних, що належать постачальникам послуг, для розміщення їх систем.

14.3. Технології віртуалізації

Операційні системи (ОС) відокремлюють програми від апаратних засобів. ОС створюють "абстракцію" деталей апаратних ресурсів програми. Віртуалізація відокремлює ОС від апаратного забезпечення.

Хмарні провайдери пропонують послуги, які можуть динамічно надавати сервери за потребою. Віртуалізація сервера використовує переваги простоюючих ресурсів на фізичній машині та консолідує кілька віртуальних серверів на одній машині. Це також дозволяє на одній апаратній платформі використовувати декілька операційних систем. Наприклад, на рис.14.2 оригінальні вісім виділених серверів були об'єднані у два сервери за допомогою гіпервізорів для підтримки декількох віртуальних екземплярів ОС.

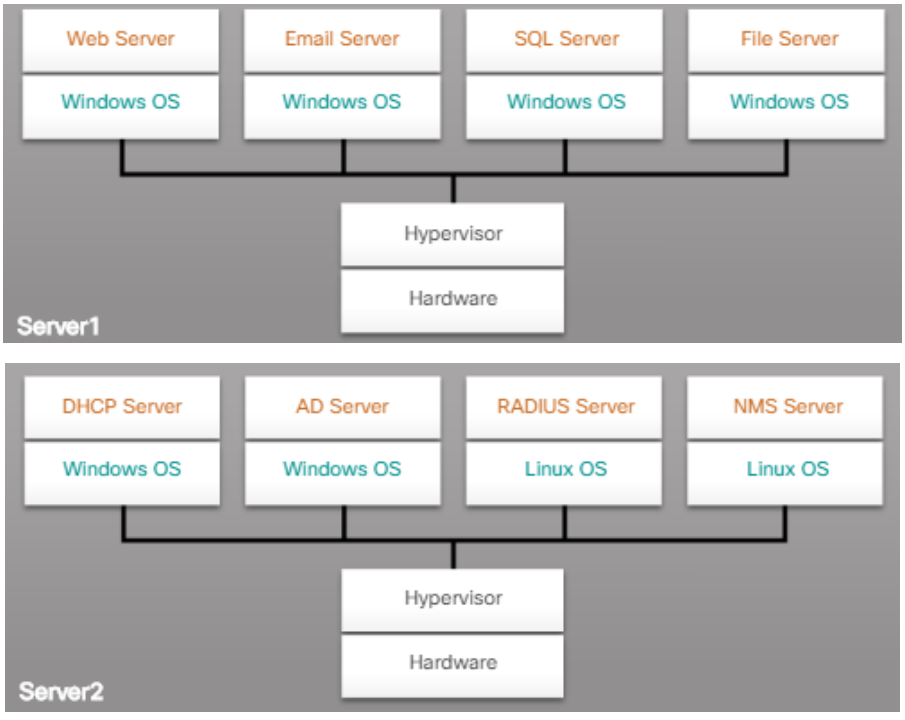

Рис. 14.2. Встановлення гіпервізора ОС [1]

Гіпервізор – це програма, прошивка або апаратне забезпечення, яке додає шар абстракції поверх реального фізичного обладнання.

Шар абстракції використовується для створення віртуальних машин, які мають доступ до всіх апаратних засобів фізичної машини, таких як процесори, пам'ять, дискові контролери та NIC. Кожна з цих віртуальних машин працює з окремою операційною системою. Завдяки віртуалізації підприємства можуть консолідувати кількість серверів, якими вони володіють та працюють. Наприклад, не рідкість 100 фізичних серверів консолідуватись як віртуальні машини на вершині 10 фізичних серверів за допомогою гіпервізорів.

Використання віртуалізації зазвичай включає надмірність для захисту від єдиної точки відмови. Надлишок може бути реалізований різними способами. Якщо гіпервізор не працює, VM можна перезапустити на іншому гіпервізорі. Крім того, однакова VM може працювати одночасно на двох гіпервізорах, копіюючи між собою інструкції оперативної пам'яті та процесора. Якщо один гіпервізор виходить з ладу, VM продовжує працювати на іншому гіпервізорі.

14.4. Шари абстракції

Щоб пояснити, як працює віртуалізація, корисно використовувати шари абстракції в комп'ютерних архітектурах. Шари абстракції (програми, ОС, обладнання) також використовуються еталонною моделлю OSI, щоб допомогти описати мережеві протоколи (рис.14.3).

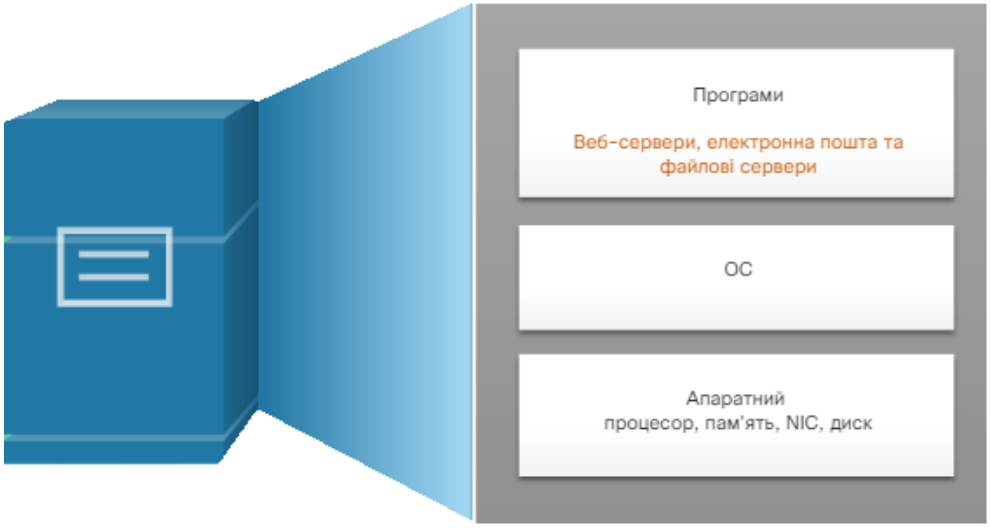

Рис. 14.3. Шари абстракції комп'ютерної системи (програми, ОС, обладнання) [1]

На кожному з цих шарів абстракції деякий тип коду програмування використовується як інтерфейс між шаром внизу та шаром вгорі. Наприклад, мова програмування С часто використовується для програмування мікропрограмного забезпечення, яке здійснює доступ до обладнання.

Приклад віртуалізації показаний на рис.14.4. Гіпервізор встановлюється між прошивкою та ОС. Гіпервізор може підтримувати кілька примірників ОС.

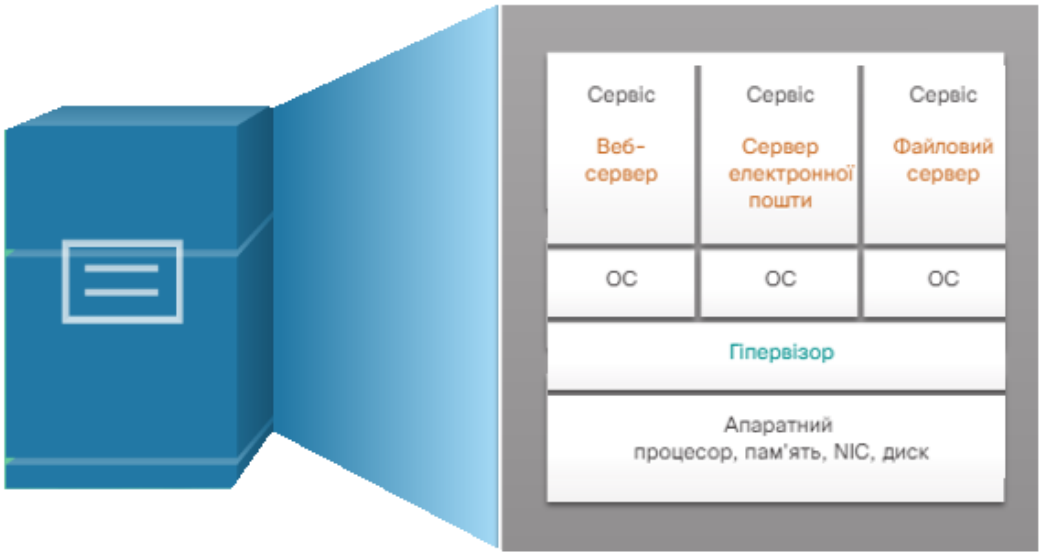

Рис. 14.3. Шари абстракції віртуальної архітектури [1]

14. 5. Гіпервізори

Гіпервізор – це програмне забезпечення, яке створює та запускає екземпляри VM. Комп'ютер, на якому гіпервізор підтримує одну або декілька VMs, є хост-машиною. Існує два наступні типи гіпервізорів.

• **Гіпервізор типу 1** – є підходом «голого металу», оскільки гіпервізор встановлюється безпосередньо на апаратному забезпеченні, (рис.14.4). Гіпервізори типу 1 зазвичай використовуються на серверах підприємства.

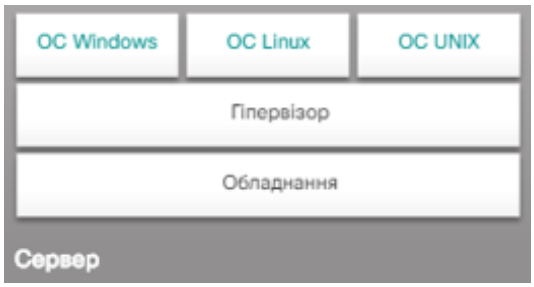

Рис. 14.4. Гіпервізор типу 1 [1]

• **Гіпервізор 2 типу** – є підходом «розміщення». Гіпервізор типу 2 додає додатковий шар абстракції. Це відбувається тому, що гіпервізор – це програма, що працює на ОС фізичного хоста, а додаткові екземпляри ОС встановлюються в гіпервізорі (рис. 14.5).

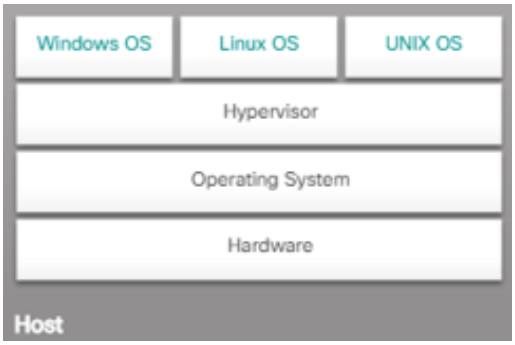

Рис. 14.5. Гіпервізор типу 2 [1]

14.6. Контейнерна технологія виконання програмного коду на сервері

Гіпервізори дозволяють кожній віртуальній машині мати власну операційну систему під час спільного використання одного і того ж обладнання. Ця конфігурація є марною, якщо операційні системи, що використовуються у віртуальних машинах, такі самі, як операційна система, що працює на хост-комп'ютері. Контейнери вирішують цю проблему.

Контейнер – це спеціалізована «віртуальна область», де програми можуть працювати незалежно одна від одної під час спільного використання однієї ОС та обладнання. З точки зору програми, це єдиний додаток, що працює на комп’ютері. Обмінюючись операційною системою хосту, більшість програмних ресурсів повторно використовуються, що оптимізує роботу.

Контейнеру потрібна лише необхідна частина операційної системи, системні ресурси та будь-які програми та бібліотеки, необхідні для запуску програми. Це дозволяє серверу підтримувати набагато більше контейнерів, ніж це могла зробити віртуальна машина в будь-який момент (рис. 14.6).

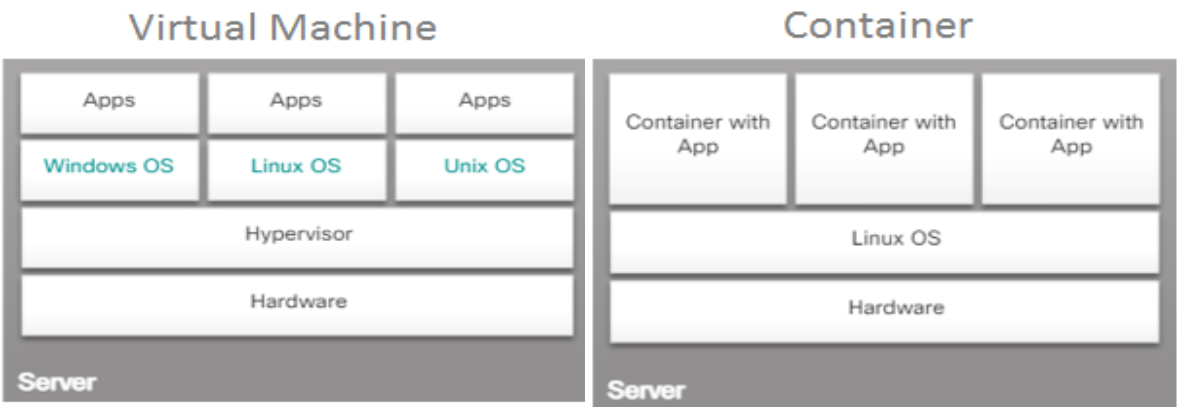

Рис. 14.6. Структура контейнеру порівняно з традиційним гіпервізором [1]

Більше програм можна запустити на сервері за допомогою контейнерної технології, контейнери вимагають, щоб операційна система віртуальної машини була такою ж, як і хост-комп'ютер. Якщо є потреба в декількох операційних системах, слід використовувати гіпервізори.

Центри обробки даних також можуть використовувати віртуалізацію для зниження витрат і розширення пропозицій хмарних постачальників.

Amazon Web Services (AWS) – хмарний постачальник послуг, який пропонує обчислювальні ресурси та послуги за потребою в хмарі. Це означає, що можна працювати на вимогу на одному або декількох віртуальних серверах на AWS, коли вони потрібні протягом певного часу.

За допомогою AWS можна зберігати дані, розміщувати веб-сайти та веб-додатки, розміщувати систему управління навчанням (Learning Management System, LMS) та обробляти великі дані, створені IoT. Це приклад IaaS (Infrastructure as a Service), де створення обчислювальної інфраструктури перетворюється на придбання послуги. Машинне навчання Amazon дозволяє розробникам створювати програми для виявлення шахрайства, прогнозування попиту, цільового маркетингу та прогнозування кліків. Алгоритми машинного навчання Amazon створюють моделі машинного навчання (ML), знаходячи шаблони в існуючих даних. Потім ці моделі обробляють нові дані та генерують прогнози. Одним із можливих застосувань моделі ML є передбачення, з якою ймовірністю клієнт придбає певний товар, виходячи з його минулої поведінки. Існує багато хмарних провайдерів. У Північній Америці вони включають

Microsoft Azure та Google Cloud. У Європі деякі хмарні постачальники – Aruba Cloud, UpCloud та CenturyLink.

Віртуалізація пам'яті поєднує в собі фізичне зберігання з декількох мережевих пристроїв зберігання даних, що видається одним пристроєм зберігання даних. Пристрій зберігання даних керується з центральної консолі. Віртуалізація пам'яті робить резервне копіювання, архівування та відновлення простішими та швидшими. Віртуалізація зберігання реалізується за допомогою програмного забезпечення, або з апаратними та програмними гібридними пристроями. Переваги віртуалізації зберігання включають збільшення ємності зберігання, автоматизоване управління, скорочення часу простою та спрощення оновлення.

Віртуальна мережа (NV) – це створення віртуальних мереж у рамках віртуалізованої інфраструктури. Процес поєднує апаратні та програмні мережеві ресурси та мережеві функціональні можливості в єдину, створену на основі програмного забезпечення адміністративну структуру. Ця сутність є віртуальною мережею. Віртуалізація мережі поєднує мережеві ресурси, розділяючи пропускну здатність на канали. Кожен канал не залежить від інших, і призначається певному серверу або пристрою. Кожен канал незалежно захищений. Кожен абонент мережі має спільний доступ до всіх ресурсів цієї мережі. Для віртуалізації мережі функція площини управління знімається з кожного пристрою і виконується централізованим контролером (рис. 14.7).

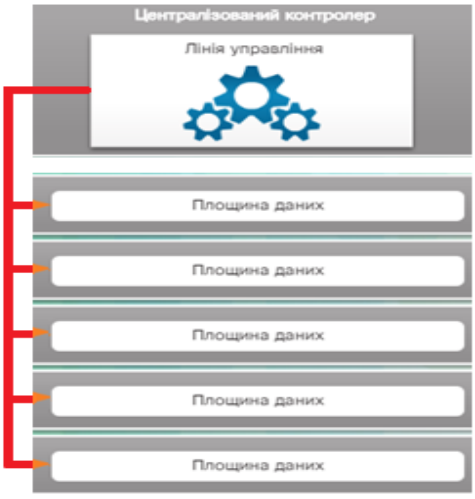

Рис. 14.7. Централізований контролер та лінія управління [1]

Централізований контролер повідомляє функції площини управління кожному пристрою. Кожен пристрій тепер може зосередитися на передачі даних, тоді як централізований контролер управляє потоком даних, підвищує безпеку та забезпечує ланцюжок послуг. Дані центри часто є середовищем, де живуть багато користувачів. NV може надавати окремі віртуальні мережі різним клієнтам у віртуальному середовищі. Ця віртуальна мережа повністю відокремлена від інших мережевих ресурсів, трафік розділяється на зони або контейнери, щоб не змішуватися з іншими ресурсами.

14.7. Інжиніринг даних

Інжиніринг даних включає інформаційну систему, де інформація (дані) збирається або формується, обробляється, зберігається, поширюється та аналізується. Можливість аналізу даних зазвичай виконується за допомогою бази даних та системи управління базами даних (СУБД). Інженерія даних та аналіз даних корисні для будь-якого бізнесу чи організації, яка хоче спрямувати свої ресурси на основі змістовної інформації та статистики.

Реляційна база даних та мова програмування структурованої мови запитів (SQL) є основою системи управління реляційними базами даних (RDMS). SQL – мова програмування, розроблена для збору інформації з реляційних баз даних за допомогою систем управління реляційними базами даних (RDBMS). Реляційні системи баз даних, такі як MySQL, Microsoft SQL Server, Oracle та IBM DB2, є найпопулярнішими системами управління базами даних. У RDBMS може бути кілька користувачів з багатьма транзакціями баз даних. Модель транзакцій Atomic, Consistent, Isolated and Durable (ACID) визначає, як транзакції бази даних підтримують цілісність даних та переживають збої.

На початку 2000-х, поява Web 2.0, електронної комерції та таких компаній, як Google, дали зрозуміти, що реляційні бази даних не змогли задовольнити об'єм та швидкість пошукових запитів у мережі. Для задоволення цього попиту Google розробила розподілену файлову систему Google (GFS), алгоритм розподіленої паралельної обробки MapReduce та розподілену базу даних NoSQL BigTable. У

2004 році Джеффрі Дін та Санджай Гемамат з Google опублікували статтю "Спрощена обробка даних на великих кластерах", яка надихнула двох програмістів Дуга Кейтінга та Майка Кафарелла створити Apache Hadoop. Після цього підхід MapReduce став основою для розробки екосистеми Hadoop і бази даних HBase, а також бази даних NoSQL з ключовими значеннями Amazon Dynamo.

Бази даних NoSQL можуть використовувати підхід для зберігання ключових значень замість підходу на основі реляційної таблиці. Інші бази даних NoSQL зберігають дані як структуровані документи у форматах XML або JSON. Бази даних NoSQL значно швидші, ніж реляційні, і можуть імпортувати неструктуровані дані. Бази даних NoSQL розроблені для масштабування по горизонталі, а це означає, що ємність зберігання та управління може бути збільшена просто додаванням інших машин до кластеру. До найпопулярніших систем NoSQL належать MongoDB, Couchbase, Riak, Memcached, Redis, CouchDB, Hazelcast, Apache Cassandra, HBase та Dynamo, які є всіма програмними продуктами з відкритим кодом.

Усі ці технології стали вирішенням проблеми Big Data. Дані настільки великі, швидкі або різноманітні, що ним неможливо керувати одним комп'ютером. З цієї причини прийнято називати ці програмні рішення «технологіями великих даних». Насправді проблеми з великими даними не можуть бути зведені до жодної технології, але повинні включати нові та старі технології.

З появою IoT та Big Data з’являються нові категорії робочих місць та коригування існуючих робочих місць.

Оцифровка бізнесу створює багато доступних бізнес-даних. Щоб отримати необхідні дані, важливо мати можливість задати правильне питання правильним чином. Бізнес-аналітик – це людина, яка може вивчати бізнес чи галузь, а потім сформулювати конкретне питання. Бізнес-аналітики – це експерти з даних, які співпрацюють із зацікавленими сторонами компанії, щоб визначити проблемне питання. Це питання потім переформулюється в конкретну проблему даних.

Потім бізнес-аналітики створюють звіти про бізнес-аналітику різних типів для зацікавлених сторін компанії.

Аналітики даних запитують та обробляють дані, надають звіти, узагальнюють та візуалізують дані. Вони використовують існуючі інструменти та методи для вирішення проблеми, допомагають розуміти конкретні запити за допомогою спеціальних звітів та графіків. Аналітикам даних необхідно зрозуміти основні статистичні принципи, процес очищення різних типів даних, візуалізацію даних та дослідницький аналіз даних. Деякі інструменти та програми, які допомагають аналітикам даних виконувати свою роботу, – це SAS, Rapid Miner та мови програмування, такі як R або Python.

Аналітик даних бере вихідні дані і перетворює їх на змістовну інформацію. Такі дослідники застосовують статистику, машинне навчання та аналітичні підходи для відповіді на критичні питання бізнесу. Наука даних – це вже існуюче поле, яке розширилося завдяки IoT та Big Data.

Аналітики даних повинні інтерпретувати та надавати результати своїх висновків методами візуалізації, будуючи програми для наукових даних. Вони працюють з наборами даних різного розміру та форм та запускають алгоритми на великих наборах даних. Деякі інструменти, які допомагають вченим даним виконувати свою роботу, – це Python, R, Scala, Apache Spark, Hadoop, data mining tools and algorithms, machine learning, statistics.

Жодна з трьох деталей, описаних вище, не може існувати без інженера даних. Інженери даних створюють інфраструктуру, яка підтримує Big Data. Вони проектують та будують платформу, на якій усі ці дані зберігаються та обробляються. Інженери даних також керують усіма цими даними. Вони забезпечують доступність даних для науковців та аналітиків.

Інженери даних можуть інтегрувати дані з різних джерел та проводити очищення даних. Однак, оскільки інженери даних розробляють в першу чергу інфраструктуру Big Data для своєї компанії, вони, як правило, не повинні знати машинного навчання чи аналітики. Деякі інструменти та програми, якими інженери даних регулярно користуються, є Hadoop, MapReduce, Hive, Pig,

MySQL, MongoDB, Cassandra, потокова передача даних, NoSQL, SQL та програмування. У деяких середовищах можливе перекриття між аналітиками даних, науковцями та інженерами даних. Коли IoT зростає і великі дані стають ще більш поширеними, посадові інструкції можуть також змінюватися.

Висновок до лекції 14

Віртуалізований центр обробки даних підтримує Big Data та аналітику. За допомогою туманних обчислень дані можуть оброблятися майже відразу після їх створення. Центри обробки даних – це централізовані місця, що містять велику кількість обчислювального та мережевого обладнання. Віртуалізація відокремлює ОС від апаратного забезпечення. Віртуалізація сховища поєднує фізичне сховище з декількох мережевих пристроїв зберігання даних у сховище, що здається єдиним запам'ятовуючим пристроєм.

Мережева віртуалізація (NV) – це створення віртуальних мереж у межах віртуалізованої інфраструктури. Інженерія даних включає в себе збір, оброблення, зберігання, розподіл та аналіз інформації.

Питання для закріплення

1. Які архітектурні моделі інженерії Big Data ви знаєте?
2. Для чого використовуються центри обробки даних та хмарні обчислення?
3. Для чого використовуються технології віртуалізації?
4. Що таке гіпервізори?
5. Що таке контейнери?
6. У чому полягає контейнерна технологія виконання програмного коду на сервері?
7. Що таке інжиніринг даних?

Список рекомендованої літератури

1. IoT Fundamentals: Big Data & Analytics // Електронний ресурс. Режим доступу: https://www.netacad.com/courses/iot/big-data-analytics

2. Virtualization Technology // Електронний ресурс. Режим доступу: https://www.sciencedirect.com/topics/computer-science/virtualization-technology

3. What is virtualization technology // Електронний ресурс. Режим доступу: https://pandorafms.com/blog/virtualization-technology/

4. Containerization // Електронний ресурс. Режим доступу: https://www.ibm.com/cloud/learn/containerization

5. What are containers (container-based virtualization or containerization) // Електронний ресурс. Режим доступу: https://searchitoperations.techtarget.com/definition/container-containerization-or-container-based-virtualization