

Міністерство освіти і науки України



Грід-системи та технології хмарних обчислень
Конспект лекцій для студентів спеціальності
8.05010101 “Інформаційні управляючі системи та технології”
денної форми навчання

Луцьк 2016

УДК 004.42(07)

До друку _____ Голова Навчально-методичної ради Луцького НТУ
(підпис)

Електронна копія друкованого видання передана для внесення в репозитарій Луцького НТУ
_____ директор бібліотеки.
(підпис)

Затверджено Навчально-методичною радою Луцького НТУ,
протокол № _ від «> листопада 2016 року.

Рекомендовано до видання Навчально-методичною радою факультету КНІТ Луцького НТУ,
протокол № __ від «> листопада 2016 року.

_____ Голова навчально-методичної ради факультет КНІТ
(підпис)

Розглянуто і схвалено на засіданні кафедри КТ Луцького НТУ,
протокол № _ від «> _____ 2016 року.

Укладач: _____ А.А.Ящук, к.т.н. ас, П.В.Саварин, ас Луцького НТУ
(підпис)

Рецензент: _____ Г.А.Герасимчук, к.т.н., доцент, декан ФКНІТ Луцького НТУ
(підпис)

Відповідальний

за випуск: _____ О.О. Герасимчук, к.т.н., доцент. Луцького НТУ
(підпис)

Грід-системи та технології хмарних обчислень [Текст] : конспект лекцій для спеціальності 8.05010101 “Інформаційні управляючі системи та технології” денної форми навчання / уклад. А.А. Ящук, П.В.Саварин – Луцьк : Луцький НТУ, 2016. – 28 с.

Видання містить необхідний теоретичний матеріал, з основ ґрід-систем та технологій хмарних обчислень, який дасть можливість студентам засвоїти навчальний матеріал з мінімальними витратами часу.

Призначене для студентів спеціальності 8.05010101 “Інформаційні управляючі системи та технології”

© А.А.Ящук, П.В.Саварин 2016

Вступ

«Грід-системи та технології хмарних обчислень» належить до циклу професійної та практичної підготовки студентів спеціальності 8.05010101 “Інформаційні управляючі системи та технології” освітньо-кваліфікаційного рівня «магістр».

Мета дисципліни: вивчення основ Грід-технологій, які використовуються для забезпечення наукових досліджень, та набуття практичних навичок використання грід-технологій для вирішення науково-практичних завдань.

Завдання: набуття навичок і вмінь практичного застосування грід-систем.

У результаті вивчення дисципліни студенти повинні:

знати:

- основи грід-технологій, які дозволяють об’єднати обчислювальні ресурси та ресурси зберігання даних в єдину систему;
- архітектуру грід-систем, які використовуються в Україні;
- принципи функціонування основних складових частин грід-систем, технологію підготовки завдань для використання грід-середовища;
- технології хмарних обчислень;

уміти:

- вибирати і використовувати проміжне програмне забезпечення для вирішення науково-практичних завдань, адаптувати пакети прикладних програм до середовища грід,
- використовувати вхідні мови для опису завдання і даних, відслідковувати та коригувати хід обчислювального процесу під час числового експерименту чи процесу моделювання.
- застосовувати на практиці технології хмарних обчислень.

Лекція 1. Консолідація ІТ інфраструктури

Консолідація – це об'єднання обчислювальних ресурсів або структур управління в єдиному центрі.

Аналіз міжнародного досвіду дозволяє сьогодні говорити про чітку тенденцію до консолідації ІТ-ресурсів корпорацій. Саме вона здатна суттєво зменшити витрати на ІТ. Заощаджені ж кошти можна направити на підвищення якості наявних інформаційних послуг та впровадження нових. Крім оптимізації витрат на ІТ, консолідація ІТ-ресурсів дозволяє поліпшити керованість підприємств за рахунок більш актуальною і повної інформації про їх функціонування. Зазвичай кажуть про консолідацію:

– **серверів** – переміщення децентралізованих, додатків, розподілених на різних серверах компанії, в один кластер централізованих гомогенних серверів;

– **систем зберігання** – спільне використання централізованої системи зберігання даних декількома гетерогенними вузлами;

– **додатків** – розміщення декількох додатків на одному хості.

При цьому можна виділити два базових типи консолідації - фізичну і логічну. Фізична консолідація уявляє географічне переміщення серверів на єдину площадку (в центр даних), а логічна - централізацію управління.

Переміщення комп'ютерів в єдиний центр обробки даних дозволяють забезпечити комфортні умови для обладнання та технічного персоналу, а також збільшити ступінь фізичного захисту серверів. Крім того, в центрі обробки даних можна використовувати більш продуктивне і високоякісне обладнання, яке економічно неефективно встановлювати в кожному підрозділі. Створюючи центри обробки даних, можна знизити витрати на технічну підтримку і управління найважливішими серверами підприємства. Вдалим прикладом обладнання, яке може успішно вирішити завдання консолідації обчислювальних ресурсів в організаціях будь-якого рівня є блейд-системи, а також і системи та мережі зберігання даних.

Очевидна перевага цього рішення в тому, що спрощується виділення персоналу підтримки і його робота з розгортання та управління системами, знижується ступінь дублювання досвідчених кадрів. Централізація також полегшує використання стандартизованих конфігурацій і процесів управління, створення рентабельних систем резервного копіювання для відновлення даних після збою і підтримки зв'язності бізнесу. Спрощується і вирішення питань організації високоякісного контролю за станом навколишнього середовища та забезпечення фізичного захисту. Може бути поліпшена і мережева безпека, оскільки сервери опиняються під захистом єдиного, централізовано керованого брандмауера.

Логічний тип консолідації уявляє перебудову системи управління ІТ-інфраструктури. Це необхідно як для збільшення масштабованості і керованості складної розподіленої обчислювальної системи, так і для об'єднання сегментів корпоративної мережі. Логічна консолідація забезпечує

введення централізованого управління та уніфікацію роботи з ресурсами компанії на основі відкритих стандартів. У результаті з'являється можливість створення глобальних інформаційних служб підприємства - каталогу LDAP, корпоративного порталу або ERP-системи, що в кінцевому підсумку дозволить поліпшити керуваність підприємства за рахунок більш актуальною і повної інформації про його функціонуванні.

Логічна консолідація додатків призводить до централізації управління критичними для бізнесу системами і додатками. Переваги логічної консолідації очевидні: в першу чергу це вивільнення апаратних ресурсів, які можна використовувати на інших ділянках інформаційної системи. По-друге, більш проста і логічна структура управління IT-інфраструктурою робить її більш гнучкою і пристосованою для майбутніх змін.

Сценарій гомогенної консолідації передбачає перенесення одного масштабного додатку, який раніше виконував на декількох серверах, на один, більш потужний. Як приклад такої операції можна навести бази даних, які часто нарощують екстенсивним шляхом у міру зростання обсягу оброблюваної інформації. Об'єднання даних і додатків на одному сервері помітно прискорює процеси обробки і пошуку, а також підвищує рівень цілісності.

Гетерогенна консолідація за змістом схожа з гомогенною, але в цьому випадку об'єднанню підлягають різні додатки. Наприклад, кілька примірників Exchange Server і SQL Server, раніше запускалися на окремих комп'ютерах, можуть бути зведені на єдиній машині. Переваги гетерогенної консолідації – зростаюча масштабованість сервісів і більш повне задіяння системних ресурсів.

Як відзначають фахівці з хмарних технологій консолідація IT-інфраструктури є першим кроком до "хмари". Щоб перейти до використання хмарних технологій, компаніям необхідно спочатку вирішити завдання неконсолідованій IT-інфраструктури. "Без консолідації неможливо побудувати ефективне процесно-орієнтоване управління, оскільки відсутня єдина точка надання сервісів".

Аналізуючи історію розвитку інформаційних технологій та сучасні тенденції можна зробити висновок, що еволюційний виток IT, що почався разом з епохою мейнфреймів понад п'ятдесят років тому, замкнувся – разом з хмарами ми повернулися до централізації ресурсів, але на цей раз не на рівні мейнфреймів з їх терміналами а на новому технологічному рівні.

Виступаючи на конференції, присвяченій проблемам сучасних процесорів, професор Масачусетського технологічного інституту Ананд Агарвал сказав: "Процесор – це транзистор сучасності". Новий рівень відрізняється тим, що тут також збираються мейнфрейми, але віртуальні, і не з окремих транзисторів, як півстоліття тому, а з цілих процесорів або цілком з комп'ютерів. На зорі IT численні компанії та організації "ліпили" власні комп'ютери з дискретних компонентів, монтуючи їх на саморобних друкованих платах - кожна організація робила свою машину, і ні про яку

стандартизацію чи уніфікацію мови не могло бути. На порозі другого десятиліття XXI століття ситуація повторюється – точно так само з, серверів, комп'ютерів, різноманітного мережевого обладнання, збираються зовнішні та приватні хмари. Одночасно спостерігається та ж сама технологічна роз'єднаність і відсутність уніфікації: Microsoft, Google, IBM, Aptana, Heroku, Rackspace, Ning, Salesforce будують глобальні мейнфрейми, а хтось під власні потреби створює приватні хмари, які є тими ж мейнфреймами, але меншого масштабу. Залишається припустити, що попереду винахід інтегральної схеми і мікропроцесора.

Ключові терміни:

Мейнфрейм – це головний *комп'ютер* обчислювального центру з великим об'ємом внутрішньої і зовнішньої пам'яті.

Блейд-сервер – комп'ютерний *сервер* з компонентами, винесеними і узагальненими в кошику для зменшення займаного простору.

Система зберігання даних (СЗД) - це програмно-апаратне рішення по організації надійного зберігання інформаційних ресурсів та надання до них гарантованого доступу.

SAN - це високошвидкісна комутована *мережа передачі даних*, яка об'єднує сервери, робочі станції, дискові сховища і стрічкові бібліотеки. *Обмін даними* відбувається по протоколу FibreChannel, оптимізованому для швидкої гарантованої передачі повідомлень і дозволяє передавати інформацію *навідстань* від декількох метрів до сотень кілометрів.

Консолідація – це *об'єднання* обчислювальних ресурсів або структур управління в єдиному центрі.

Лекція 2. Технології віртуалізації

Згідно зі статистикою середній рівень завантаження процесорних потужностей у серверів під управлінням Windows не перевищує 10%, у Unix-систем цей показник краще, але тим не менше в середньому не перевищує 20%. Низька ефективність використання серверів пояснюється широко застосовуваним з початку 90-х років підходом "один додаток – один сервер", тобто кожен раз для розгортання нової програми компанія застосовує новий сервер. Очевидно, що на практиці це означає швидке збільшення серверного парку і як наслідок - зростання витрат на його адміністрування, енергоспоживання та охолодження, а також потребу в додаткових приміщеннях для установки все нових серверів і придбанні ліцензій на серверну ОС.

Віртуалізація ресурсів фізичного сервера дозволяє гнучко розподіляти їх між додатками, кожен з яких при цьому "бачить" тільки призначені йому ресурси і "вважає", що йому виділено окремий сервер, тобто в даному випадку реалізується підхід "один сервер – кілька додатків", але без зниження продуктивності, доступності та безпеки серверних додатків. Крім того,

рішення віртуалізації дають можливість запускати в розділах різні ОС за допомогою емуляції їх системних викликів до апаратних ресурсів сервера.

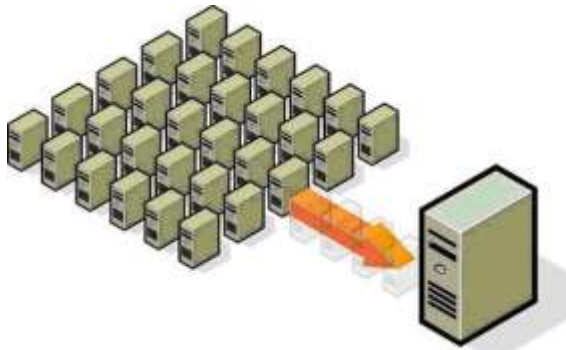


Рис. 2.1. Віртуалізація як запуск на одному фізичному комп'ютері декількох віртуальних комп'ютерів

В основі віртуалізації лежить можливість одного комп'ютера виконувати роботу декількох комп'ютерів завдяки розподілу його ресурсів на декілька середовищ. За допомогою віртуальних серверів і віртуальних настільних комп'ютерів можна розмістити кілька ОС і кілька додатків в єдиному місцезнаходженні. Таким чином, фізичні та географічні обмеження перестають мати якусь значення. Крім енергозбереження та скорочення витрат завдяки більш ефективному використанню апаратних ресурсів, віртуальна інфраструктура забезпечує високий рівень доступності ресурсів, більш ефективну систему управління, підвищену безпеку і вдосконалену систему відновлення в критичних ситуаціях.

У широкому розумінні поняття віртуалізації являє собою приховування справжньої реалізації якого-небудь процесу або об'єкта від істинного його подання для того, хто ним користується. Продуктом віртуалізації є щось зручне для використання, насправді, має більш складну або зовсім іншу структуру, відмінну від тієї, яка сприймається при роботі з об'єктом. Іншими словами, відбувається відділення подання від реалізації чого-небудь. Віртуалізація покликана абстрагувати програмне забезпечення від апаратної частини.

У комп'ютерних технологіях під терміном "віртуалізація" зазвичай розуміється абстракція обчислювальних ресурсів і надання користувачеві системи, яка "інкапсулює" (приховує в собі) власну реалізацію. Простіше кажучи, користувач працює з зручним для себе представленням об'єкта, і для нього не має значення, як об'єкт влаштований в дійсності.

Зараз можливість запуску декількох віртуальних машин на одній фізичній викликає великий інтерес серед комп'ютерних фахівців, не тільки

тому, що це підвищує гнучкість ІТ-інфраструктури, але й тому, що віртуалізація, насправді, дозволяє економити гроші.

Історія розвитку технологій віртуалізації налічує більше сорока років. Компанія ІВМ була першою, хто задумався про створення віртуальних середовищ для різних користувальницьких завдань, тоді ще в мейнфреймах. У 60-х роках минулого століття віртуалізація представляла чисто науковий інтерес і була оригінальним рішенням для ізоляції комп'ютерних систем в рамках одного фізичного комп'ютера. Після появи персональних комп'ютерів інтерес до віртуалізації дещо послабився зважаючи бурхливий розвиток операційних систем, які пред'являли адекватні вимоги до апаратного забезпечення того часу. Однак бурхливе зростання апаратних потужностей комп'ютерів наприкінці дев'яностих років минулого століття змусив ІТ-спільноту знову згадати про технології віртуалізації програмних платформ.

У 1999 р компанія VMware представила технологію віртуалізації систем на базі x86 в якості ефективного засобу, здатного перетворити системи на базі x86 в єдину апаратну інфраструктуру загального користування та призначення, що забезпечує повну ізоляцію, мобільність і широкий вибір ОС для прикладних середовищ. Компанія VMware була однією з перших, хто зробив серйозну ставку виключно на віртуалізацію. Як показав час, це виявилось абсолютно виправданим. Сьогодні VMware пропонує комплексну віртуалізаційну платформу четвертого покоління VMware vSphere 4, яка включає ресурси як для окремого ПК, так і для центру обробки даних. Ключовим компонентом цього програмного комплексу є гіпервізор VMware ESXServer. Пізніше в "битву" за місце в цьому модному напрямку розвитку інформаційних технологій включилися такі компанії як Parallels (раніше SWsoft), Oracle (Sun Microsystems), Citrix Systems (XenSource).

Корпорація Microsoft вийшла на ринок засобів віртуалізації в 2003 р з придбанням компанії Connectix, випустивши свій перший продукт Virtual PC для настільних ПК. З тих пір вона послідовно нарощувала спектр пропозицій у цій галузі і на сьогодні майже завершила формування віртуалізаційних платформи, до складу якої входять такі рішення як Windows 2008Server R2 з компонентом Hyper-V, Microsoft Application Virtualization (App-v), Microsoft Virtual Desktop Infrastructure (VDI), Remote Desktop Services, System Center Virtual MachineManager.

На сьогоднішній день постачальники технологій віртуалізації пропонують надійні і легкокеровані платформи, а ринок цих технологій переживає справжній бум. За оцінками провідних експертів, зараз віртуалізація входить до трійки найбільш перспективних комп'ютерних технологій. Багато експертів пророкують, що до 2015 року близько половини всіх комп'ютерних систем будуть віртуальними.

Підвищений інтерес до технологій віртуалізації в даний час не випадковий. Обчислювальна потужність нинішніх процесорів швидко зростає, і питання навіть не в тому, на що цю міць витратити, а в тому, що

сучасна "мода" на двоядерні і багатоядерні системи, проникла вже і в персональні комп'ютери (ноутбуки та десктопи), як не можна краще дозволяє реалізувати багатий потенціал ідей віртуалізації операційних систем і додатків, виводячи зручність користування комп'ютером на новий якісний рівень. Технології віртуалізації стають одним з ключових компонентів (у тому числі, і маркетингових) в найновіших і майбутніх процесорах Intel і AMD, в операційних системах від Microsoft і ряду інших компаній.

Переваги віртуалізації:

1. Ефективне використання обчислювальних ресурсів. Замість 3х, а то 10 серверів, завантажених на 5-20% можна використовувати один, що використовується на 50-70%.

2. Скорочення витрат на інфраструктуру. Віртуалізація дозволяє скоротити кількість серверів і пов'язаного з ними ІТ-обладнання в інформаційному центрі.

3. Зниження витрат на програмне забезпечення. Деякі виробники програмного забезпечення ввели окремі схеми ліцензування спеціально для віртуальних середовищ.

4. Підвищення гнучкості і швидкості реагування системи. Віртуалізація пропонує новий метод управління ІТ-інфраструктурою і допомагає ІТ-адміністраторам затрачати менше часу на виконання повторюваних завдань – наприклад, на ініціацію, настройку, відстежування і технічне обслуговування.

5. Несумісні додатки можуть працювати на одному комп'ютері.

6. Підвищення доступності додатків і забезпечення безперервності роботи підприємства. Завдяки надійній системі резервного копіювання та міграції віртуальних середовищ цілком без перерв в обслуговуванні ви зможете скоротити періоди планового простою і забезпечити швидке відновлення системи в критичних ситуаціях.

7. Можливості легкої архівації. Оскільки жорсткий диск віртуальної машини зазвичай представляється у вигляді файлу певного формату, розташованого на якому-небудь фізичному носії, віртуалізація дає можливість простого копіювання цього файлу на резервний носій як засіб архівування і резервного копіювання всієї віртуальної машини цілком.

8. Підвищення керованості інфраструктури – використання централізованого управління віртуальною інфраструктурою дозволяє скоротити час на адміністрування серверів, забезпечує балансування навантаження і "живу" міграцію віртуальних машин.

Віртуальною машиною будемо називати програмну або апаратне середовище, яке приховує справжню реалізацію якого-небудь процесу або об'єкта від його видимого уявлення .

Віртуальна машина – це повністю ізольований програмний контейнер, який працює з власною ОС і додатками, подібно фізичному комп'ютеру. Віртуальна машина діє так само, як фізичний комп'ютер, і містить власні віртуальні (тобто програмні) ОЗУ, жорсткий диск і мережевий адаптер .

ОС не може розрізнити віртуальну і фізичну машини. Те ж саме можна сказати про додатки та інші комп'ютери в мережі. Навіть сама *віртуальна машина* вважає себе "справжнім" комп'ютером. Але незважаючи на це віртуальні машини складаються виключно з програмних компонентів і не включають обладнання. Це дає їм ряд унікальних переваг над фізичним обладнанням.



Рис. 2.2. Віртуальна машина

Розглянемо основні особливості віртуальних машин більш детально:

1. **Сумісність.** Віртуальні машини, як правило, сумісні з усіма стандартними комп'ютерами. Як і фізичний комп'ютер, віртуальна машина працює під управлінням власної гостьової операційної системи і виконує власні додатки. Вона також містить усі компоненти, стандартні для фізичного комп'ютера (материнську плату, відеокарту, мережевий контролер і т.д.). Тому віртуальні машини повністю сумісні з усіма стандартними операційними системами, програмами та драйверами пристроїв.

2. **Ізольованість.** Віртуальні машини повністю ізольовані одна від одної, як якщо б вони були фізичними комп'ютерами.

3. **Інкапсуляція.** Віртуальні машини повністю інкапсулюють обчислювальне середовище. Віртуальна машина являє собою програмний контейнер, що зв'язує "інкапсулюючий" повний комплект віртуальних апаратних ресурсів, а також ОС і всі її додатки в програмному пакеті.

4. **Незалежність від устаткування.** Віртуальні машини повністю незалежні від базового фізичного обладнання, на якому вони працюють.

Основні різновиди віртуалізації:

- віртуалізація серверів (повна віртуалізація і паравіртуалізація)
- віртуалізація на рівні операційних систем,
- віртуалізація додатків,
- віртуалізація подань.

Лекція 3. Концепція Грід. Відмінності від традиційних розподілених систем

Останні п'ятнадцять років є роками зародження та розвитку нового напрямку в інформаційних технологіях, назву якому (як традиційно вважається) дали у 1998 році Я. Фостер та К. Кессельман – «Грід» (англ. «Grid»). Grid як засіб сумісного використання обчислювальних потужностей та сховищ даних дозволяє вийти за межі простого обміну даними між комп'ютерами і, зрештою, перетворити їх глобальну мережу на свого роду гігантський віртуальний комп'ютер, доступний у режимі віддаленого доступу з будь-якої точки, незалежно від місця розташування користувача.

Ідея використовувати мережу суперкомп'ютерів для вирішення задач, мабуть, зародилася набагато раніше (спроби робилися з 60-х років XX століття), однак зараз вона набула завершеної форми «концепції Grid». Традиційно, причетними до розвитку Grid -обчислень вважають фізиків-ядерщиків - і дотепер їх потреба в обробці колосальних об'ємів дослідних даних є рушійною силою для реалізації програм по впровадженню Grid, згадати хоча б діяльність Європейського центру ядерних досліджень (CERN). Однак Grid має потенційно велику кількість й інших областей застосування, оскільки пропонує універсальний підхід до розв'язку проблеми нестачі обчислювальних ресурсів – адже очевидно, що в загальному випадку мережа суперкомп'ютерів є спроможною вирішити складніші задачі, ніж кожен з її складових вузлів окремо.

Найближче за смыслом до «Грід» є поняття «power grid» – мережа електроживлення, розподілений ресурс загального користування, коли кожен може легко під'єднатися через розетку і використовувати стільки електроенергії, скільки йому потрібно. Аналогічно користувачі з допомогою Grid отримують можливість прямого підключення до віддаленої обчислювальної мережі, не цікавлячись, звідки беруться потрібні для роботи обчислювальні ресурси й дані, які для цього використовуються лінії передачі, паролі чи протоколи тощо. При цьому аналогом інфраструктури електричних мереж (ліній електропередачі, підстанцій, трансформаторів, диспетчерських пунктів та ін.) виступає Grid –інфраструктура з програмним Grid забезпеченням (ПГЗ), з допомогою якого виконується „віртуалізація” ресурсів. Розвиток ПГЗ починався від базових засобів, що підтримують дистанційний доступ до ресурсів, пройшов стадію окремих систем, їх пакетів і привів до створення платформ – взаємоузгоджених наборів засобів, здатних дати комплексне рішення завдання обслуговування Grid-інфраструктури виробничого призначення.

Концепція Grid базується на наступних основоположних моментах:

- швидке та постійне збільшення продуктивності мікропроцесорів масового виробництва (сучасний персональний комп'ютер на базі процесора може зрівнятися за швидкістю обчислень із суперкомп'ютерами 10-літньої давнини);
- поява швидкісних оптоволоконних ліній зв'язку - сьогодні в розвинених країнах базові лінії зв'язку в мережі Інтернет мають пропускну здатність 10 Гб/с і вище, а підключення до мережі багатьох наукових організацій відбувається на швидкості в 1-2 Гб/с;
- феномен Інтернету, глобалізація процесу обміну інформацією й інтеграції світової економіки;
- безперервне вдосконалювання технологій і засобів інформаційної безпеки.

По суті, грід є «надбудовою» над Інтернетом. Її основне призначення – організація розподілених обчислень для рішення серйозних задач науки й технології, які вимагають більших обчислювальних ресурсів, а саме, потужності комп'ютерів, ресурсів зберігання даних, часу обчислень. На відміну від безструктурної мережі Інтернет, грід – чітко впорядкована система. Певною мірою грід можна умовно назвати обчислювальним Інтернетом. Користувач, підключаючись до грід, одержує доступ до потужності тисяч машин, на яких він може здійснювати обчислення й зберігати як величезні масиви даних, так і інформацію, отриману в результаті їхньої обробки. У цій мережі приділяється першорядна увага проблемам безпеки – адж анонімність, зручна при спілкуванні в Інтернеті, може стати надзвичайно небезпечною при роботі з науковими або практичними даними. Запитуючи яку-небудь інформацію в глобальній базі даних грід, користувач одержить вичерпну відповідь на питання про її достовірність, повноту й доступність.

Необхідно зазначити відмінність грід-технології від технології паралельних обчислень. Основними перешкодами для здійснення нетривіальних паралельних обчислень в грід-середовищі є нестабільність, погана передбачуваність часу відгуку на запит. Причому це пов'язано не лише з тим, що в комп'ютерних мережах інформаційні пакети проходять через безліч мережевих пристроїв, але і з відмінностями в протоколах зв'язку, які використовуються у "зовнішніх" комп'ютерних мережах і для міжпроцесорного обміну всередині суперкомп'ютерів. Це не дозволяє ефективно організувати паралельні обчислення з інтенсивним обміном інформацією між процесорами, які виконують окремі підзадачі в грід-середовищі.

Грід-технологія не є технологією паралельних обчислень, вона призначена для віддаленого запуску окремих завдань на територіально розподілених ресурсах. Тому якщо громіздке завдання може бути розбите на ряд маленьких, незалежних (які не обмінюються жодними даними) частин, - грід-технологія виявляється особливо ефективним і відносно дешевим

рішенням. Навпаки, суперкомп'ютери виявляються для таких обчислень не виправдано дорогими і неефективними. Існують і гібридні проекти, метою яких є досягнення максимальної швидкості обчислень за рахунок глобального розподілу цих обчислень між суперкомп'ютерами - при цьому грід координує використання різних суперкомп'ютерів, а власне нетривіальне розпаралелювання відбувається всередині суперкомп'ютера.

Лекція 4. Вимоги та характеристики грід-системи

Є два основних критерії, що відрізняють Грід-системи від інших систем, що забезпечують роздільний доступ до ресурсів:

1. Грід-система координує розрізнені ресурси. Ресурси не мають загального центру керування, а грід-система займається координацією їхнього використання, наприклад, балансуванням навантаження. Тому проста система керування ресурсами кластера не є системою грід, тому що здійснює централізоване керування всіма вузлами даного кластера, маючи до них повний доступ. Грід-системи мають лише обмежений доступ до ресурсів, що залежить від політики того адміністративного домена, у якому цей ресурс перебуває.

2. Грід-система будується тільки на базі стандартних і відкритих протоколів, сервісів і інтерфейсів.

Додаткові властивості, які повинні мати грід-системи:

- гнучкість (можливість забезпечення поділюваного доступу потенційно до будь-яких видів ресурсів);
- масштабованість (працездатність при збільшенні чи зменшенні складу системи);
- гнучка і потужна підсистема безпеки (стійкість до атак злоумисників, забезпечення конфіденційності);
- можливість контролю над ресурсами (застосування локальних і глобальних політик і квот);
- гарантії якості обслуговування;
- можливість одночасної, скоординованої роботи з декількома ресурсами.

Найчастіше реалізації Грід-систем забезпечують роботу з такими типами ресурсів:

- обчислювальні ресурси – окремі комп'ютери, кластери;
- ресурси зберігання даних – диски і дискові масиви, стрічки, системи масового зберігання даних;
- мережні ресурси;
- програмне забезпечення – яке-небудь спеціалізоване ПЗ.

Є різниця між технологією грід і реалізаціями грід-систем. Технологія грід – це найбільш загальні і універсальні аспекти, однакові для будь-якої системи (архітектура, протоколи, інтерфейси, сервіси).

Нижче коротко перераховуються та пояснюються основні властивості грід, до яких відносять:

– *спільне використання розподілених ресурсів*. Воно відкриває можливості з співробітництва, яких були б важко досягти іншими засобами. Водночас виникають питання «справедливості» розподілу ресурсів, створення та управління віртуальними організаціями (ВО).

– *об'єднання потужностей*. Таким чином будується система з сумарною потенційною обчислювальною потужністю, що перевершує потужності її складових, і при цьому досягається більш ефективне використання апаратних засобів (зменшується простоювання). Постають жорсткі вимоги до каналів зв'язку.

– *віртуалізація*. Включає в себе такі поняття, як: приховування (маскування) від користувача складності апаратної та програмної реалізації системи та її складових, географічних відстаней між вузлами, приналежності вузлів різним організаціям, створення ілюзії роботи з «віртуальним суперкомп'ютером».

– *неоднорідність* (гетерогенність). Типовий Grid складається з множини різнорідних апаратних засобів та різноманітного програмного забезпечення (і має успішно функціонувати в таких умовах). Децентралізоване управління. Немає одного єдиного «власника» всієї системи, що вимагає використання механізмів розподіленого управління.

– *інтероперабельність*. Функціональна сумісність роботи різних компонентів Grid та навіть різних Grid -інфраструктур базується на стандартизації інтерфейсів. Підходи, що не враховують стандартів, є мало перспективними.

– *прозорість доступу*. Grid має надавати доступ до ресурсів системи користувачам, не зважаючи на конкретну топологію мережі чи локальну реалізацію механізмів доступу до тих чи інших вузлів та їх ресурсів.

– *масштабованість*. Grid має забезпечувати механізми включення нових джерел ресурсів, користувачів, сховищ даних тощо без впливу на існуючих учасників: Grid повинен мати здатність динамічно реконфігуруватись. Безпека. Одна з головних вимог до Grid - системи – безпека доступу до ресурсів, що зумовлює обмежений набір дозволених операцій у авторизованих користувачів та програм.

Лекція 5. Стандарти побудови архітектури грід

У останні декілька років технологія Grid еволюціонувала від ретельно конфігурованої інфраструктури, яка підтримувала виконання обмеженого числа додатків категорії Grant Challenge на високопродуктивній апаратурі, до динамічного середовища, розвиток якого направляє міжнародне співтовариство. У міру становлення технології Grid реальністю до процесу її

розвитку все більш залучається індустрія. Участь комерційних організацій прискорює розробку надійного програмного забезпечення, що підтримує середовища Grid за межами академічних лабораторій. У свою чергу, це впливає як на архітектуру Grid, так і на пов'язані з нею протоколи і стандарти. Пристосування до архітектури гід технології Web-сервісів принесло істотну користь та привело до появи дещо фрагментованого середовища розробки. Розробники програмного забезпечення і Grid-сервісів добиваються відповідності угодам і стандартам, широко поширеним в їх співтоваристві. Проте по різних політичних і технічних причинах є декілька точок зору, що змагаються, щодо того, як слід реалізувати архітектуру, і на які стандарти потрібно спиратися. Це суперництво гальмує розробників програмного забезпечення Grid, оскільки вони не упевнені, що майбутні стандарти включатимуть ті, що використовуються сьогодні. Основною організацією по стандартизації Grid є Global Grid Forum (GGF www.ggf.org). Крім того, роботи по стандартизації ведуться в Organization for the Advancement of Structured Information Standards (OASIS www.oasis.org), World Wide Consortium (W3C www.w3c.org), Distributed Management Task Force (DMTF www.dmtf.org), Web Services Interoperability Organization (WS-I www.ws-i.org), Internet2 (www.internet2.edu) і Liberty Alliance (www.projectliberty.org). Найбільш важливим стандартом, покликаним визначити загальну, стандартну і відкриту архітектуру Grid, є стандарт Open Grid Services Architecture (OGSA), що розвивається GGF. У березні 2004 р. була випущена перша версія стандарту (OGSA 1.0), а в червні 2005 р. вийшла друга версія стандарту. OGSA є сервіс-орієнтованою архітектурою, в якій специфікується набір розподілених обчислювальних патернів, що реалізуються з використанням Web-сервісів. Стандарт призначається для визначення всіх основних сервісів, які можуть використовуватися в додатках e-business або e-science, включаючи управління роботами і ресурсами, комунікації і безпеку. Робота по специфікації інтерфейсів сервісів, семантики, протоколів і інших технічних деталей надана різним робочим групам усередині GGF і іншим організаціям по стандартизації Grid.

Перша конкретизація OGSA була здійснена в документі Open Grid Services Infrastructure (OGSI), випущеному в липні 2003 р. Цей документ базувався на понятті Grid-сервіса, розширенні Web-сервіса, в якому забезпечувався стандартний набір механізмів для управління станом. У OGSI 1.0 визначається набір принципів і розширень для використання WSDL і XML Schema при організації Web-сервісів з підтримкою стану. Критики OGSI відзначали ряд проблем в цьому стандарті: дуже великий об'єм; потреба в розширенні стандартного WSDL; дуже сильна об'єктна орієнтованість. Це привело до виникнення руху за визначення альтернативної інфраструктури Grid, заснованої на чистих специфікаціях Web-сервісів.

20 січня 2004 р. HP, IBM, Fujitsu і Globus Alliance оголосили про випуск WS-Resource Framework (WSRF www.globus.org/wsrp). Цей документ складається з набору специфікацій для виразу зв'язку між ресурсами, що

володіють станами, і Web-сервісами. У специфікаціях визначаються конкретні формати повідомлень і зв'язані визначення на XML. Остаточні результати були передані двом новим технічним комітетам OASIS: WS-Resource Framework (WSRF) TC і WS-Notification (WSN) TC. WSRF TC відповідає за стандартизацію специфікацій WS-Resource Lifetime (визначаються способи управління життєвим циклом ресурсу і специфікуються Web-сервіси для ліквідації ресурсу); WS-Resource Properties (визначаються способи запиту і модифікації ресурсів, що описуються XML-документами Resource Property); WS-ServiceGroup (визначаються способи представлення і управління колекціями Web-сервісів і/або WS-ресурсами); WS-BaseFaults (визначається базовий XML-тип, використовуваний при обміні повідомленнями в Web-сервісах для інформування про збої). WSN TC займається стандартизацією трьох специфікацій, що відносяться до інтерфейсів Web-сервісів: WS-BaseNotification (асинхронне сповіщення, що включає інтерфейси виробника і споживача); WS-BrokeredNotification (асинхронне сповіщення); WS-Topics (організація і категоризація тем для підписки).

Деякі організації, що ведуть або планують грід-проекти, користуються альтернативними специфікаціями, що включають Basic Profile (BP1.0) від WS-I, Web Services Grid Application Framework (WS-GAF, North-East Regional e-Science Centre www.neresc.ac.uk/ws-gaf) і WS-I+ (Open Middleware Infrastructure Institute www.omii.ac.uk). Специфікація BP1.0 була опублікована в квітні 2004 р. і містила керівництво по використанню SOAP, WSDL і UDDI. У WS-GAF пропонується підхід, відмінний від OGSF, до розширення функціональності Web-сервісів для задоволення потреб Grid-додатків. У WS-I+ указуються існуючі стандарти, які є потенційно сумісними із стандартами Grid, що розвиваються. Фактичним стандартом безпеки в Grid є Grid Security Infrastructure (GSI <http://forge.gridforum.org/projects/gsi-wg>). У двох нових проектах досліджуються альтернативні рішення, які можуть вплинути на стандарти GSI. У проектах GridShib (<http://grid.ncsa.uiuc.edu/GridShib>) і ESP-GRID (<http://e-science.ox.ac.uk/oesc/projects>) створені нові механізми і стратегії розподіленої аутентифікації, що дозволяють віртуальним організаціям в Grid інтегруватися з традиційною інфраструктурою корпоративної безпеки.

Лекція 6. Розвиток грід-систем

Перші проекти грід були спробами зв'язати суперобчислювальні вузли (сайти); у той час цей підхід був відомий як *метаобчислення*. Метакомп'ютер – це багатопроцесорна система, роль системної шини в якій виконує Інтернет, що зв'язує безліч вузлів, що мають власні процесори, оперативну і зовнішню пам'ять, пристрої введення/виведення.

Початком цього періоду вважається старт проекту CASA в 1989 році одним з декількох американських іспитових стендів Gigabit. На початку та

середині дев'яностих років відзначають появу перших метаобчислень, або середовищ ґрід. Як правило, мета цих ранніх метаобчислювальних проєктів полягала в тому, щоб забезпечити ряд високопродуктивних додатків обчислювальними ресурсами. Поперед цієї технології йшли два представницьких проєкти – FAFNER та I-WAY. Ці проєкти відрізняються різними шляхами реалізації, але для ефективної роботи вони повинні були подолати багато подібних перешкод, такі як зв'язок, керування ресурсами та віддаленими даними. Ці два проєкти також спробували вирішити проблеми метаобчислення, при цьому FAFNER був здатний до виконання на будь-якій робочій станції, що мала більше ніж 4 мегабайти пам'яті, I-WAY був засобом об'єднання ресурсів величезних американських суперобчислювальних центрів.

FAFNER мав недоліки, які зараз у Grid-системах відсутні. Наприклад, кожний клієнт повинен був компілювати, встановлювати і виконати демон FAFNER, щоб здійснювати факторизації. Індивідуальні обчислювальні завдання були нездатні об'єднуватися один з одним, або з батьківським Web-сервером. Аналогічно, I-WAY мав багато таких властивостей, які сьогодні здаються недоречними, наприклад, сервіси I-WAY повинні були бути встановлені окремо на кожному вузлі. Незважаючи на вищезгадані особливості, FAFNER і I-WAY були досить інноваційними та успішними проєктами.

Подальший розвиток і узагальнення ідей *метакомп'ютингу* на більш ширше коло обчислювальних ресурсів і завдань/додатків вимагав невідкладного рішення широкого кола проблем, пов'язаних з передачею даних, забезпеченням безпеки, управлінням завданнями, доступом до даним, пошуком ресурсів, доступом до них та інше. Таку ґрід-систему можна розглядати як представника ґрід-систем другого покоління, яким притаманні три основних властивості:

- *різноманітність*: ґрід використовує різноманітні ресурси, які є різнорідними за походженням, і могли б охопити численні адміністративні домени через потенційно глобальний простір;

- *масштабованість*: ґрід міг би зростати від декількох ресурсів до тисяч. Це підіймає проблему потенційного зниження рівня продуктивності по мірі росту та поширення ґрід. Отже, повинні бути розроблені додатки, які спроможні використовувати наявні ресурси, а також значно покращені засоби аутентифікації користувачів;

- *адаптивність*: в ґрід відмова ресурсу є скоріше правилом, ніж виключенням. Фактично, з дуже багатьма ресурсами в ґрід ймовірність деякого збою ресурсу досить висока. Менеджери ресурсу або додатки повинні динамічно адаптуватися таким чином, щоб вони могли витягти максимальну продуктивність із доступних ресурсів і сервісів.

Як відмічалось, ПГЗ (або *програмане забезпечення проміжного шару*) використовується для того, щоб приховати різнорідну природу ґрід та створити користувачам і додаткам однорідне середовище, забезпечуючи ряд

стандартизованих інтерфейсів і безліч сервісів. ПГЗ перебуває між операційною системою і додатками, забезпечуючи додатки безліч сервісами, необхідними для їх коректного функціонування у розподілених гетерогенних середовищах.

Грід-інфраструктура може складатися з будь-яких видів мережевих ресурсів, починаючи від комп'ютерів (обчислювальних ресурсів) до сховищ даних і спеціальних наукових приладів. Традиційно грід-додатки потребують величезні за масштабом об'єми даних і обчислень. Виникла потреба в створенні нового типу програмного забезпечення грід систем, яке на відміну від традиційних клієнт-серверних систем, де ресурси управляються і облік користувачів ведеться централізовано в одному довірчому домені.

Друге покоління базового програмного забезпечення для грід у своєму розвитку перейшло від ранніх систем типу Globus-GT1 і Legion, спеціалізованих для конкретних потреб великих і високопродуктивних додатків, до більш універсальних і відкритих середовищ, таких, як Globus-GT3 і Avaki.

Однак друге покоління ПГЗ не забезпечило повну функціональну сумісність створеного грід програмного забезпечення, що є важливою умовою реалізації великомасштабних обчислень. Оскільки подальші рішення грід продовжували досліджуватися, стали очевидними інші аспекти розробки грід.

Дві ключових особливості грід-систем третього покоління – наголос на прийнятті *сервісно-орієнтованої моделі* та зростаюча *увага до метаданих*. Фактично сервісно-орієнтований підхід припускає, що гнучке застосування ресурсів грід у додатках грід вимагатиме інформацію про функціональні можливості, характеристики та інтерфейси різних компонентів, і ця інформація повинна бути погодженою, щоб можна було її обробляти комп'ютером.

Були введені нові терміни «розподілене співробітництво» і «віртуальна організація». В третьому поколінні сформувався більш цілісне уявлення про виконання грід-обчислень, і можна сказати, що справа йдеться скоріше про інфраструктуру для е-Науки, ніж про поліпшення уже розробленої існуючої технології. Очікуване використання обчислювальних засобів з масовим паралелізмом – тільки частина картини, що намалювалася, існує також величезна кількість користувачів, отже розподілені обчислення не були прерогативою тільки комп'ютерів з масовим паралелізмом.

Грід може забезпечити підтримку віддаленого спілкування вчених в реальному часі. Особливо важливою є інфраструктура для підтримки розподілених ресурсів – тут є багато ключових сервісів: безпека, масштабування і управління, реєстрація і пошук, та інтерфейси Web-сервісів, орієнтовані на повідомлення, для забезпечення могутніх механізмів співробітництва. Всі головні грід-сервіси і інфраструктура забезпечують співпрацю і є вкрай значущими для суспільства.

Існує безліч прикладів, які ілюструють вражаюче зростання кількості наукових даних, що генеруються. Як приклад можна розглянути завдання контролю робочого стану промислового устаткування. Британська e-Science програма профінансувала проект DAME (Distributed Aircraft Maintenance Environment) – розподілене оточення технічного обслуговування авіації [67], який присвячений аналізу тих даних, що генеруються аеродвигунами компанії Rolls Royce. Вважається, що зараз використовується багато тисяч двигунів виробництва Rolls Royce. Для прикладу, кожен трансатлантичний переліт здійснений двигуном, генерує біля гігабайта даних – від датчиків тиску, температури і коливання. Мета проекту – передати невелику частинку цих первинних даних для аналізу і порівняння з даними, які зберігаються в трьох центрах в різних точках миру. Шляхом отримання перших результатів Rolls Royce сподівається збільшити період між плановими техобслуговуваннями, таким чином збільшивши рентабельність двигунів. Датчики двигунів генеруватимуть безліч петабайт даних за рік але в реальному часі повинні будуть ухвалюватися рішення про те скільки даних аналізувати, скільки передавати для подальшого аналізу і скільки архівувати.

До найбільш вагомих здобутків грид-систем третього покоління, отриманих до тепер, можна віднести:

- злиття грид-технологій і технологій *Web-сервісів*, формування грид сервісу як спеціального розширення Web-сервісу шляхом підтримки *екземплярів грид-сервісу*, що мають *стан* і, можливо, обмежений *час життя*; З кожним екземпляром грид-сервісу пов'язані *дані сервісу*, тобто інформація, структурована у вигляді набору іменованих XML-елементів, що типізуються (service data elements, SDE

- забезпечення *інтероперабельності* різних реалізацій сервісів, визначення стандартизованих інтерфейсів сервісів (OGSI), визначення протоколу(-ів) для виклику певного інтерфейсу, домовленість про стандартний набір підтримуваних протоколів, при чому для кожного із стандартних інтерфейсів визначений набір елементів даних сервісу, які повинні підтримуватися будь-яким екземпляром сервісу, що реалізовує даний інтерфейс;

- запропонування спільними зусиллями GT, IBM та інших компаній набору специфікацій під ім'ям WS-Resource Framework (WSRF), який спирається на ту ж архітектуру OGSA, на загальновизнані стандарти Web-сервісів, при чому зберігається багато елементів OGSI, але використовується інша термінологія і розширюються можливості OGSI;

- створення програмному грид забезпечення Globus Toolkit GT4 як відкритої реалізації WSRF і засоби розробки клієнтських і серверних додатків на мовах Java, C++ і Python; при цьому GT4 не повністю сумісний з попереднім GT3.

Лекція 7. Види хмарних обчислень

З поняттям хмарних обчислень часто пов'язують такі сервіс-надають (Everything as a service) технології, як:

- "Інфраструктура як сервіс" ("Infrastructure as a Service" або "IaaS")
- "Платформа як сервіс" ("Platform as a Service", "PaaS")
- "Програмне забезпечення як сервіс" ("Software as a Service" або "SaaS").

Розглянемо кожну з цих технологій докладніше.

Інфраструктура як сервіс (IaaS)

IaaS - це надання комп'ютерної інфраструктури як послуги на основі концепції хмарних обчислень.

IaaS складається з трьох основних компонентів:

1. Апаратні засоби (сервери, системи зберігання даних, клієнтські системи, мережеве обладнання)
2. Операційні системи та системне ПЗ (засоби віртуалізації, автоматизації, основні засоби управління ресурсами)
3. Сполучна ПО (наприклад, для управління системами)



Рис. 7.1. Компоненти хмарної інфраструктури

IaaS заснована на технології віртуалізації, що дозволяє користувачеві обладнання ділити його на частини, які відповідають поточним потребам бізнесу, тим самим збільшуючи ефективність використання наявних обчислювальних потужностей. Користувач (компанія або розробник ПЗ) повинен буде оплачувати всього лише реально необхідні йому для роботи серверний час, дисковий простір, мережеву пропускну спроможність та інші ресурси. Крім того, IaaS надає в розпорядження клієнта весь набір функцій управління в одній інтегрованій платформі.

IaaS позбавляє підприємства від необхідності підтримки складних інфраструктур центрів обробки даних, клієнтських і мережевих інфраструктур, а також дозволяє зменшити пов'язані з цим капітальні витрати і поточні витрати. Крім того, можна отримати додаткову економію, при наданні послуги в рамках інфраструктури спільного використання.

Першопрохідцями в IaaS вважається компанія Amazon, які на сьогоднішній день пропонують два основних IaaS-продукту: EC2 (**Elastic**

Compute Cloud) і S3 (**Simple Storage Service**). EC2 являє собою Хен-хостинг зі статичними VPS-характеристиками, що не розширюються на льоту (хоча багато подібні сервіси вже надають т.зв. auto scaling). Сховище S3 має інтерфейс *WebDAV* і підтримує роботу з багатьма відомими мовами програмування.

Серед інших інфра-сервісних компаній можна відзначити:

GoGrid має дуже зручний інтерфейс для управління VPS, а також *cloud storage* з підтримкою протоколів SCP, FTP, SAMBA /CIFS, RSYNC, причому розмір сховища масштабується на льоту. Незабаром розробники обіцяють додати управління за допомогою API.

Enomally являє собою рішення для розгортання та управління віртуальними додатками в хмарі, при цьому управління послугами здійснюється через браузер. Приємним доповненням є автоматичне масштабування віртуальних машин під поточну навантаження, а також автобалансування навантаження. Серед підтримуваних віртуальних архітектур підтримуються Linux, Windows, Solaris і BSD Guests. Для віртуалізації застосовують не тільки Xen, а й KVM, а також VMware.

Eucalyptus являє собою програмний комплекс з відкритим кодом для реалізації *cloud computing* на кластерних системах. В даний час інтерфейс сумісний з Amazon EC2, але заявлена підтримка та інших.

Лекція 8. Веб-служби в хмарі

Оскільки технології мігрують від традиційної локальної моделі в нову модель хмари, сервісні пропозиції розвиваються практично щодня. Пропозиції веб-служб часто мають багато спільних характеристик. Часто від клієнта потрібні лише мінімальні *витрати* для отримання послуги. Масштабованість передбачається для кожного з типів пропозицій, але це не завжди необхідно. Багато хто з "хмарних" вендорів ще працюють над використанням масштабованості, тому що їх користувачі поки не потребують даному виді послуг. Нарешті, пристрій і незалежність місця розташування дозволяє користувачам отримати *доступ* до систем незалежно від того, де вони знаходяться або які пристрої використовують.

Інфраструктура як Сервіс (Infrastructure-as-a-Service, IaaS) – надання комп'ютерної інфраструктури (як правило, це платформи віртуалізації) як сервісу. IaaS суттєво підсилює технологію, послуги та вкладення в центри обробки даних, щоб надати це як послугу клієнтам. На відміну від традиційного аутсорсингу, який вимагає належної старанності, нескінченних переговорів і складних, довгих контрактів, IaaS сфокусована навколо моделі надання послуг, яка забезпечує зумовлену, стандартизовану інфраструктуру, безумовно оптимізовану під потреби клієнта. Спрощені пропозиції роботи і вибір рівня сервісного обслуговування полегшує клієнтові вибір рішення з певним набором основних експлуатаційних

характеристик. Як правило, постачальники надають компоненти наступних рівнів:

- апаратне забезпечення (як правило, Грід з масивною горизонтальною масштабованістю);
- комп'ютерна мережа (включаючи маршрутизатори, брандмауери, балансування навантаження і т.д.);
- підключення Інтернет;
- платформа віртуалізації для того, щоб запускати віртуальні машини;
- угоди сервісного обслуговування;
- інструменти обліку обчислень.

Замість покупки простору в центрах обробки даних, серверів, програмного забезпечення, мережного обладнання, і т.д., клієнти IaaS по суті орендують ресурси, які знаходяться на стороні обслуговуючих постачальників послуг IaaS. Оплата за надання послуг зазвичай проводиться щомісяця. Клієнт платить тільки за спожиті ресурси.

Обчислення на вимогу набувають все більшої популярності серед підприємств. Обчислювальні ресурси, які обслуговують користувача веб сайти, стають все менше і менше, в той час, як доступні ресурси постачальників послуг постійно зростають. Модель на вимогу розвинулася, щоб подолати проблему того, як ефективно задовольнити вагається вимогам системи до ресурсів. Попит на обчислювальні ресурси може істотно змінюватися за досить короткі проміжки часу, і підтримка ресурсів достатніх, щоб задовольнити піковим вимогам може бути дорогим. Технічно переускладнення рішення може бути настільки ж несприятливою, як ситуація, коли підприємство скорочує витрати, підтримуючи тільки мінімальні обчислювальні ресурси. Такі поняття як кластерні обчислення, грід обчислення і т.д., можуть здаватися дуже подібними поняттю обчислень на вимогу, але краще їх зрозуміти можна, якщо думати про них, як стандартні блоки, які розвивалися протягом довгого часу, щоб досягти сучасної моделі хмарних обчислень, яку ми використовуємо сьогодні.

Amazon

Розглянемо один із прикладів – Amazon's Elastic Compute Cloud (Amazon EC2). *Amazon EC2* - веб-служба, яка забезпечує обчислювальні потужності значного розміру в хмарі.

Amazon EC2 представляє обчислювальне середовище, дозволяючи клієнтам використовувати веб інтерфейс, для отримання та управління послугами, необхідними для запуску одного або більше примірників операційної системи. *Amazon EC2* дозволяє користувачам збільшувати або зменшувати продуктивність за кілька хвилин. Користувачі можуть запускати єдиний екземпляр, сотні примірників або навіть тисячі екземплярів служб одночасно. Усім цим керують за допомогою API веб-служби, додаток може автоматично масштабувати себе вгору або вниз, залежно від його потреб.

Amazon EC2 працює в з'єднанні з іншими веб-службами Amazon. Наприклад, *Amazon Simple Storage Service (Amazon S3)*, *Amazon SimpleDB*, *Amazon Simple Queue Service (Amazon SQS)* і *Amazon CloudFront* все інтегровані, щоб забезпечити повне рішення для обчислень, обробки запитів і зберігання між широким діапазоном додатків.

Amazon S3 забезпечує інтерфейс веб-служб, який дозволяє користувачам зберігати і відновлювати будь-який обсяг даних через Інтернет в будь-який час, будь-де.

Amazon SimpleDB – інший веб сервіс, розроблений для того, щоб виконувати запити на структурованих даних *Amazon Simple Storage Service (Amazon S3)* в режимі реального часу.

Amazon Simple Queue Service (Amazon SQS) – сервіс, що приймає черги повідомлень для зберігання.

Amazon CloudFront – веб-сервіс для доставки контенту (вмісту). *Amazon CloudFront* інтегрується з іншими *Amazon Web Services*. Мета сервісу – дати розробникам і підприємствам простий спосіб поширювати контент для кінцевих користувачів з низькою затримкою, високою швидкістю передачі даних, при цьому не вимагаючи ніяких зобов'язань. Сервіс не є вільним для користування.

Платформа корпорації Майкрософт Windows Azure (спочатку відома під назвою *Azure Services Platform*) – це група "хмарних" технологій, кожна з яких надає певний набір служб для розробників додатків. Платформа *Windows Azure* може бути використана як додатками, що виконуються в "хмарі", так додатками, що працюють на локальних комп'ютерах.

ЛІТЕРАТУРА

1. Петренко А.И. Вступ до Grid технологій в науці та освіті: навчальний посібник / А.И. Петренко – К.: НТУУ «КПІ», 2008. – 120 с.
2. Пономаренко В.С. Методы и модели планирования ресурсов в Grid системах / В.С. Пономаренко, С.В. Листровой, С.В. Минухин, С.В. Знахур – Х.:ВД. «ІНЖЕК», 2008.- 408 с.
3. Паклин Н.Б. Бизнес-аналитика: от данных к знаниям / Н.Б. Паклин, В.И. Орешков – Издательский дом Питер, 1-е издание, 2009 год.- 624 с.
4. Барсегян А.А. Методы и модели анализа данных: OLAP и Data Mining (+ CD-ROM) / Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. – Издательство: БХВ – Петербург, 2008 г.- 336 с.

Зміст

Вступ	3
Лекція 1 Консолідація ІТ інфраструктури.....	4
Лекція 2 Технології віртуалізації.....	6
Лекція 3 Концепція грід. Відмінності від традиційних розподілених систем.....	11
Лекція 4 Вимоги та характеристики грід-системи.....	13
Лекція 5 Стандарти побудови архітектури грід	14
Лекція 6 Розвиток грід-систем	16
Лекція 7 Види хмарних обчислень.....	19
Лекція 8 Веб-служби в хмарі.....	21
Література	24

Грід-системи та технології хмарних обчислень [Текст] : конспект лекцій для спеціальності 8.05010101 “Інформаційні управляючі системи та технології” денної форми навчання / уклад. А.А.Ящук, П.В.Саварин – Луцьк : Луцький НТУ, 2016. – 28 с.

Комп’ютерний набір
Редактор

А.А.Ящук
А.А.Ящук

Підп. до друку 2016р.
Формат 60x84/16. Папір офс. Гарнітура Таймс.
Ум. друк. арк. ____. Обл.-вид. арк. 2,5.
Тираж ____ прим. Зам. 1.

Редакційно-видавничий відділ
Луцького національного технічного університету
43018 м. Луцьк, вул. Львівська, 75
Друк – РВВ Луцького НТУ