

УДК 621.391.8

Коновалов О.Ю., к.т.н. (Київський коледж зв'язку)

## ОРГАНІЗАЦІЯ РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

**Коновалов О.Ю. Організація розподілених обчислювальних систем.** У роботі розглянуті методи обробки інформації в розподілених обчислювальних системах і організації розподілених застосувань. Виконаний огляд основних підходів організації розподілених обчислювальних систем. Приводиться опис концепції GRID-обчислень. Розглядається технологія і архітектура побудови систем хмарних обчислень.

**Ключові слова:** РОЗПОДІЛЕНА ОБЧИСЛЮВАЛЬНА СИСТЕМА, ХМАРНІ ОБЧИСЛЕННЯ, GRID-СИСТЕМА, CORBA, DCOM, SOA

**Коновалов А.Ю. Организация распределенных вычислительных систем.** В работе рассмотрены методы обработки информации в распределенных вычислительных системах и организации распределенных приложений. Выполнен обзор основных подходов организации распределенных вычислительных систем. Приводится описание концепции грид - вычислений. Рассматривается технология и архитектура построения систем облачных вычислений.

**Ключевые слова:** РАСПРЕДЕЛЕННАЯ ВИЧИСЛИТЕЛЬНАЯ СИСТЕМА, ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ, ГРИД-СИСТЕМА, CORBA, DCOM, SOA

**Konovalov O.Iu. The organizing distributed computing systems.** This article presents the main approaches to distributed information processing in Distributed Computing Systems and the organization of distributed applications. A review of the main approaches of organizing distributed computing systems. The description of the concept of GRID-computing. The technology of building systems and architecture of cloud computing.

**Keywords:** DISTRIBUTED COMPUTING SYSTEM, CLOUD COMPUTING, GRID, CORBA, DCOM, SOA

**Вступ.** Розробка розподілених обчислювальних систем базувалась на різних підходах, що залежали від існуючих програмно-апаратних рішень: концепції Веб, що була орієнтована на людину, яка повинна отримати доступ до розподіленого інформаційного простору, і технологій [1, 2] розподілених об'єктів (CORBA і DCOM), в основі яких лежала концепція створення розподілених середовищ для емуляції процесів розробки і виконання локальних застосувань, що в свою чергу, забезпечувало доступ до мережевих ресурсів. Подальший розвиток напряму залежав від нових методів і проміжного програмного забезпечення для розподілених обчислювальних систем, таких як: технології однорангових мереж (peer-to-peer або P2P) і GRID-технологій.

Метою даної статті є огляд методів організації розподілених обчислювальних систем, механізмів взаємодії обчислювальних середовищ із клієнтськими додатками, та порівняльний аналіз GRID- і хмарних обчислень.

Після початкового етапу створення і розвитку концепції розподілених середовищ революційною стала концепція P2P, що дозволило простим споживачам інформації прийняти активну участь в наданні контенту, в той час як застосування технології GRID дозволило інтегрувати великі комплекси обробки і зберігання даних, забезпечуючи їх доступність для різних урядових і наукових користувачів. Концепція GRID-обчислень була орієнтована на побудову інфраструктури, що забезпечує "обчислення на вимогу".

Якщо користуватись визначенням розподіленої системи наведеної у [3]: "Розподілена обчислювальна система (РОС) – це набір сполучених каналами зв'язку незалежних комп'ютерів, які з точки зору користувача деякого програмного забезпечення виглядають єдиним цілим", то можна виділити ряд моментів, які є ключовими для розуміння концепції розподілених обчислень, і які базуються на таких властивостях РОС як: автономність вузлів P2P і представлення системи користувачем, як єдиної структури.

Розподілена обчислювальна система є програмно-апаратним комплексом, орієнтованим на рішення певних завдань, а основною сполучною ланкою розподілених обчислювальних систем є програмне забезпечення. При цьому, кожен обчислювальний вузол є автономним елементом, а програмна складова РОС, повинна забезпечувати користувачам видимість роботи з єдиною обчислювальною системою.

Таким чином РОС не критична до складу типів пристроїв, операційних систем, апаратних платформ та алгоритмів обробки інформації.

Обчислювальні середовища, що складаються з безлічі обчислювальних систем на базі різних програмно-апаратних платформ, називаються гетерогенними, що забезпечує: *можливість* простого розширення і масштабування; *перманентну* (постійну) доступність ресурсів (навіть якщо деякі елементи РОС деякий час можуть знаходитися поза доступом); *приховання* особливостей комунікації від користувачів.

Функціонування розподілених обчислювальних систем на пряму залежить від стабільності і надійності зв'язку між її компонентами. Компоненти такої системи розподілені, тобто віддалені один від одного на значні відстані.

В основі будь-якої розподіленої обчислювальної системи лежать існуючі технологічні і апаратно-програмні рішення комп'ютерних і телекомунікаційних мереж. Організація обміну між розподіленими (територіально, адміністративно і так далі) компонентами успішно вирішуються в існуючих мережах, і, РОС використовують напрацьований досвід.

Взаємодія в обчислювальних мережах базується на протоколах. Протокол – це набір правил і угод, що описують процедуру взаємодії між компонентами системи (у тому числі і обчислювальної).

Ключовою проблемою організації обчислювальних систем є проблема вибору між централізованою і розподіленою моделями надання обчислювальних ресурсів [4].

Завданням розподілених обчислювальних систем є "гнучкий, безпечний, координований розподіл ресурсів серед динамічних наборів користувачів, організацій і ресурсів" [5]. Такі розподілені обчислювальні системи отримали термін GRID. І на цій основі було сформовано представлення концепції "хмарних обчислень", в основі якої було надання високомасштабованих віртуальних обчислювальних ресурсів кінцевому користувачеві через інтернет у вигляді послуг.

Використання розподілених обчислень у вигляді технологій GRID і хмарних обчислень сьогодні є найбільш актуальним завданням інформатизації у всьому світі. Вони застосовуються для вирішення різних завдань, їх використання стає все простіше. Розподілені обчислення стають невід'ємною частиною наукових і комерційних високопродуктивних обчислень.

Ранні проекти високопродуктивних обчислювальних систем, виявили основні проблеми, які необхідно було вирішити для розгортання стабільних розподілених обчислювальних систем. Для вирішення усіх вказаних завдань була розроблена концепція GRID – РОС, що забезпечує "гнучкий, безпечний, координований розподіл ресурсів серед динамічних наборів користувачів, організацій і ресурсів".

Основним завданням GRID була побудова інфраструктури, що забезпечує "обчислення на вимогу" (utility computing). У [6] наведено три основні вимоги, яким повинні відповідати РОС для того, щоб називатися GRID:

– *Гетерогенність*. Обчислювальне середовище GRID може складатися з множини різних ресурсів, що мають різні характеристики і параметри.

– *Масштабованість*. GRID може складатися із скільки завгодно великого числа ресурсів. І можлива ситуація, коли час виконання завдання збільшується через зменшення швидкодії при передачі даних між вузлами.

– *Пристосовність*. При роботі з GRID-системами необхідно враховувати велику вірогідність помилок при роботі з ресурсами. Середовище GRID може складатися з сотень комп'ютерів, і помилки в роботі десятка з них не повинні впливати на отримані результати.

При розгляді другого покоління розподілених обчислювальних систем не можна не торкнутися такого класу, як розподілені об'єктні системи. На початку розвитку таких систем одним з найбільш поширених методів побудови розподілених об'єктних систем була архітектура CORBA (Common Object Request Broker Architecture – загальна архітектура брокера об'єктних запитів) [7]. Друга версія специфікації стандарту CORBA, що описував стандартний протокол взаємодії об'єктно-орієнтованих систем, була опублікована консорціумом OMG (Object Management Group) у 1997-1998 роках.

Паралельно із стандартом були випущені відображення для найбільш поширених об'єктно-орієнтованих мов: C++ і Java.

До складу рішення CORBA було включено безліч сервісів, які можна було застосувати до галузей електронної комерції і науки. Також, на базі CORBA можливо було реалізувати концептуальні моделі архітектури розподіленого обчислювального середовища, оскільки вона забезпечувала взаємозв'язок з мовою UML (Unified Modeling Language - Уніфікована Мова Моделювання).

Але рішення на базі архітектури CORBA внаслідок ряду причин поступово почали відмирати. Частково це пояснюють складністю процесу розробки на основі об'єктно-орієнтованої моделі, що була запропонована CORBA, частково – високою вартістю придбання і підтримки таких систем.

Якщо розглядати модель Java у порівнянні з CORBA, то потрібно відмітити наступне: якщо для боротьби з гетерогенністю обчислювального середовища CORBA реалізує високорівневі стандарти взаємодії, модель Java використовує єдине віртуальне середовище.

Певною мірою JVM (Java Virtual Machine – Віртуальна Машина Java), в сукупності з Java-додатками і сервісами, долають проблеми гетерогенності обчислювальних систем, надаючи методи віддаленого виклику процедур за допомогою технології RMI (Remote Method Invocation – Видалений Виклик Методів).

На сьогодні, РОС відходять від традиційних понять високопродуктивних розподілених обчислень у бік розвитку віртуальної співпраці і віртуальних організацій.

Віртуальна організація – це ряд користувачів і/або організацій, об'єднаних загальними правилами колективного доступу до певних обчислювальних ресурсів [5].

Методи надання доступу до обчислювальних ресурсів стають сервісно-орієнтованими, що дозволяє гнучко використовувати одні і ті ж обчислювальні ресурси різними споживачами. На початку 2000-х років почався процес розробки наступного покоління специфікацій, покликаних вирішити проблеми ранніх стандартів розподілених об'єктних технологій за допомогою веб-сервісів і сервіс-орієнтованої архітектури (Service-Oriented Architecture – SOA).

Стандарти веб-сервісів були розроблені за ініціативою організацій, що займаються наданням віддаленого доступу до певних обчислювальних ресурсів, і закріплені консорціумом W3C. До основних стандартів розробки і функціонування веб-сервісів можна віднести: SOAP – заснований на XML протокол взаємодії веб-сервісів; WSDL (Web Services Description Language – Мова опису веб-сервісів) це методологія опису ресурсів, що надаються веб-сервісом; UDDI (Universal Description Discovery and Integration – Універсальний метод пошуку і інтеграції) – метод опису, пошуку, взаємодії і використання веб-сервісів.

Незважаючи на усі переваги технології веб-сервісів, вони не надають нових методологій і рішень для побудови широкомасштабних обчислювальних мереж. Для пошуку рішень в цьому напрямі, необхідно розглянути агентно-орієнтовану парадигму побудови РОС.

Обчислювальні мережі на основі так званих агентів – це принципово інший підхід до організації РОС. Програмний агент можна описати як автономний процес, що здатний реагувати на середовище виконання і викликати зміни в середовищі виконання, можливо, в кооперації з користувачами або іншими агентами. Основні принципи роботи агентних мереж описані в [8]: *автономність* – агенти функціонують автономно, без можливості стороннього втручання в їх внутрішній стан; *соціальна поведінка* – агенти взаємодіють один з одним за допомогою певної мови; *активність* – агенти взаємодіють з довкіллям, отримуючи певні сигнали і відповідаючи на них; *проактивність* – агенти діють цілеспрямовано.

Агентні мережі принципово пристосовані для функціонування в динамічно-змінюваному довкіллі. В цьому випадку, автономність агентів дозволяє організувати динамічне підстроювання обчислювального алгоритму під умови обчислювального середовища.

Для забезпечення функціонування такої системи, потрібна стандартизація методів взаємодії між компонентами. Для вирішення цього завдання розробляються і

стандартизуються мови взаємодії агентів (Agent Communication Languages, ACLs). Одним з найбільш відомих, являється архітектура взаємодії FIPA (Foundation for Intelligent Physical Agents, базис інтелектуальних фізичних агентів). Ця архітектура стандартизує методи взаємодії агентів і агентних систем.

Опис концепції GRID-обчислень і порівняння сучасної концепції хмарних обчислень, дають можливість відмітити, що між ними багато спільного.

Деякі дослідники вважають, що основною відмінністю хмарних обчислень від GRID є те, що хмарні обчислення, на відміну від GRID, застосовують віртуалізацію для максимізації обчислювальної потужності. Віртуалізація, за допомогою відділення логічного рівня від фізичного, вирішує безліч проблем, з якими стикаються GRID-системи.

GRID-системи забезпечують високе завантаження обчислювальних ресурсів за допомогою розподілу одного складного завдання на декілька обчислювальних вузлів, а хмарні обчислення йдуть шляхом виконання декількох завдань на одному сервері у вигляді віртуальних машин. Крім того, є особливості в основних варіантах використання GRID- і хмарних обчислень. Тоді як GRID, в основному, використовується для вирішення завдань за певний (обмежений) проміжок часу, хмарні обчислення в основному орієнтовані на надання "довгоживучих" сервісів.

Думки сходяться в одному – хмарні обчислення вирости з концепції GRID. Існує думка, що хмарні обчислення не просто перетинаються з концепцією GRID. Насправді хмари вирости з GRID-обчислень і базуються на концепції інфраструктури GRID.

Еволюція підходу полягає в тому, що замість надання "сирих" обчислювальних ресурсів і ресурсів зберігання забезпечується надання абстрактніших ресурсів у вигляді сервісів.

Хмара – це парадигма великомасштабних розподілених обчислень, заснована на ефекті масштабу, у рамках якої пул абстрактних, віртуалізованих, динамічно масштабованих обчислювальних ресурсів, ресурсів зберігання, платформ і сервісів надається за запитом зовнішнім користувачам через інтернет [9]. На сьогодні, сервіс-орієнтований підхід є стандартом при розробці розподілених обчислювальних систем.

Таким чином, можна вважати що GRID- і хмарні обчислення доповнюють один одного. Інтерфейси і протоколи GRID можуть забезпечити взаємодію між хмарними ресурсами або ж забезпечити об'єднання хмарних платформ.

Також, вищий рівень абстракції, що надається хмарними платформами, може допомогти користувачам GRID-систем в організації прозорого і зручного надання ресурсів GRID-платформ і забезпечити нові групи користувачів такими ресурсами.

З точки зору користувача, різниця між хмарними обчисленнями і GRID-обчисленнями полягатиме в наступних відмінностях. Хмарні платформи фокусуються на підході "усе як сервіс", в той час як GRID-обчислення фокусуються на проміжному програмному забезпеченні, яке надається у вигляді відкритих початкових кодів або ж у вигляді готових пакетів. В порівнянні з цим, хмарні обчислення фокусуються виключно на платному наданні інформаційних ресурсів кінцевому користувачеві. При цьому проміжне програмне забезпечення, яке дозволило б забезпечити розробку власної хмари, поки не дуже поширено.

GRID- і хмарні обчислення націлені на різні типи обчислень. Спочатку, GRID обчислення були орієнтовані на рішення наукових завдань за допомогою суперкомп'ютерних систем. Нині GRID застосовується для науково-дослідницьких завдань, вирішення яких вимагає об'єднання декількох суперкомп'ютерних платформ. З іншого боку, хмарні обчислення орієнтовані не на вирішення окремих завдань, а на перманентне надання певних сервісів кінцевим користувачам. Вони забезпечують динамічний розподіл фізичних ресурсів для задоволення змінного завантаження таких сервісів.

Також потрібно відмітити різну взаємодію з постачальниками ресурсів. GRID-обчислення ґрунтуються на понятті віртуальних організацій, що включають декілька різних окремих організацій з чіткими правилами взаємодії між ними і чіткими політиками надання програмно-апаратних ресурсів, в той час як хмарні обчислення забезпечують можливість будь-якої компанії використовувати хмарні сервіси для вирішення власних

завдань, оплачуючи тільки ті ресурси, які потрібні. GRID-платформи надають базу для розгортання обчислювальної інфраструктури, в той час як хмарні обчислення надають інтегрований підхід на усіх рівнях надання інформаційних ресурсів : SaaS, PaaS, IaaS.

Програмне забезпечення як послуга (SaaS, англ. Software-as-a-Service) – модель, в якій споживачеві надається можливість використання прикладного програмного забезпечення, що працює в хмарній інфраструктурі провайдера і доступного з різних клієнтських пристроїв або за допомогою тонкого клієнта, наприклад, з браузеру (наприклад, веб-пошта) або інтерфейс програми. Контроль і управління основною фізичною і віртуальною інфраструктурою хмари, у тому числі мережі, серверів, операційних систем, зберігання, або навіть індивідуальних можливостей додатка (за винятком обмеженого набору призначених для користувача налаштувань конфігурації додатка) здійснюється хмарним провайдером.

Платформа як послуга (PaaS, англ. Platform-as-a-Service) – модель, коли хмарний провайдер, надає споживачу можливість використання хмарної інфраструктури для розміщення базового програмного забезпечення для наступного розміщення на ній нових або існуючих застосувань (власних, розроблених на замовлення або придбаних тиражованих застосувань). До складу таких платформ входять інструментальні засоби створення, тестування і виконання прикладного програмного забезпечення – системи управління базами даних, єдине програмне забезпечення, середовища виконання мов програмування. Контроль і управління основною фізичною і віртуальною інфраструктурою хмари, у тому числі мережі, серверів, операційних систем, зберігання здійснюється хмарним провайдером, за винятком розроблених або встановлених застосувань, а також, по можливості, параметрів конфігурації середовища (платформи).

Інфраструктура як послуга (IaaS, англ. IaaS or Infrastructure-as-a-Service) надається як можливість використання хмарної інфраструктури для самостійного управління ресурсами обробки, зберігання, мереж і іншими фундаментальними обчислювальними ресурсами, наприклад, споживач може встановлювати і запускати довільне програмне забезпечення, яке може включати операційні системи, платформенне і прикладне програмне забезпечення. Споживач може контролювати операційні системи, віртуальні системи зберігання даних і встановлені застосування, а також обмежений контроль набору доступних сервісів (наприклад, міжмережевий екран, DNS). Контроль і управління основною фізичною і віртуальною інфраструктурою хмари, у тому числі мережі, серверів, типів використовуваних операційних систем, систем зберігання здійснюється хмарним провайдером.

GRID-обчислення орієнтовані на представлення різних обчислювальних ресурсів в гетерогенних обчислювальних середовищах для вирішення конкретних завдань. Таким чином, інтерфейси GRID-орієнтовані на взаємодію обчислювальних інфраструктур на фізичному рівні за допомогою API, яким може скористатися тільки професійний програміст, а хмарні обчислення розробляються так, щоб надавати інтерфейси кінцевим користувачам через веб-доступ або за допомогою API. На кожному шарі (IaaS, PaaS, SaaS) надається свій власний інтерфейс. Підвищення рівня абстракції дозволяє забезпечити застосування хмарних обчислень як на рівні окремих користувачів, так і на рівні корпоративних клієнтів.

Загалом і в цілому, GRID-обчислення забезпечують об'єднання гетерогенних обчислювальних ресурсів в єдине обчислювальне середовище. Це те, з чого починаються і на чому ґрунтуються хмарні обчислення. Хмарні обчислення забезпечують вищий рівень абстракції, надаючи обчислювальні ресурси кінцевим користувачам (будь то приватні клієнти або організації) у вигляді сервісів.

**Висновки.** В роботі представлені основні підходи до організації розподіленої обробки інформації в обчислювальних мережах і організації розподілених застосувань. Виконаний огляд основних підходів організації розподілених обчислювальних систем. Приводиться опис концепції GRID-обчислень, і виконано порівняльний аналіз технологій і архітектур побудови систем хмарних обчислень і GRID-систем.

На сьогодні розподілені обчислювальні системи міцно зайняли специфічну нішу високопродуктивних обчислень для спеціалізованих завдань і відповідних методів їх

рішення. Постійно розробляються і впроваджуються нові концепції розвитку розподілених систем, змінюється і розширюється круг вирішуваних ними завдань, спрощуються процеси організації, розробляються простіші методи використання клієнтських ресурсів.

#### Література

1. Siegel J. OMG overview: CORBA and the OMG in enterprise computing. Communications of the ACM, vol. 41, no. 10, 1998. pp. 37-43.
2. The Distributed Component Object Model (DCOM) [Електронний ресурс]. – Режим доступу : <http://www.microsoft.com/com/tech/DCOM.asp>
3. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван-Стеен. – Спб.: Питер, 2003. – 877 с.
4. King J.L. Centralized versus decentralized computing: organizational considerations and management options // ACM Computing Surveys. Vol. 15, Issue 4. 1983. P. 319-349.
5. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations // International Journal of Supercomputer Applications and High Performance Computing. 2001. Vol. 15, No 3. P. 200-222.
6. Foster I., Kesselman C. The Grid. Blueprint for a new computing infrastructure. San Francisco: Morgan Kaufman, 1999. 677 p.
7. Henning M. The Rise and Fall of CORBA // ACM Queue. Vol. 4, Num. 5. 2006. P. 28-34.
8. Jennings N. R. An agent-based approach for building complex software systems // Comms. of the ACM, 44 (4) 35-41, 2001.
9. Foster I., Zhao Y., Raicu I, Lu S. Cloud computing and grid computing 360-degree compared // Grid Computing Environments Workshop, 2008. GCE'08. pp. 1-10, 2008.

УДК 004.8.565.5

Гайдур Г.І. (Державний університет інформаційно-комунікаційних технологій)

### ВИКОРИСТАННЯ WIRELESS MESH МЕРЕЖ ДЛЯ ПІДВИЩЕННЯ НАДІЙНОСТІ СИСТЕМИ КЕРУВАННЯ У СУЧАСНИХ БЕЗПРОВОДОВИХ ТЕХНОЛОГІЯХ

**Гайдур Г.І. Використання Wireless MESH мереж для підвищення надійності системи керування у сучасних безпроводових технологіях.** В останні роки безпроводові мережі отримали широке розповсюдження. Топологія MESH використовується в інформаційних мереж 3G, WiFi, WiMax, LTE. Для таких мереж запропоновано алгоритм розрахунку оптимального шляху за приведеною пропускною спроможністю, яка дозволяє ефективно підвищити надійність MESH мереж для сучасних безпроводових технологій.

**Ключові слова:** БЕЗПРОВОДОВІ ТЕХНОЛОГІЇ, WIRELESS MESH, DCN, ГРАФ, АЛГОРИТМ

**Гайдур Г.И. Использование Wireless MESH сетей для повышения надежности системы управления в современных беспроводных технологиях.** В последние годы беспроводные сети получили широкое распространение. Топология MESH используется в информационных сетях 3G, WiFi, WiMax, LTE. Для таких сетей предложен алгоритм расчета оптимального пути по приведенной пропускной способности, которая позволяет эффективно повысить надежность MESH сетей для современных беспроводных технологий.

**Ключевые слова:** БЕСПРОВОДНЫЕ ТЕХНОЛОГИИ, WIRELESS MESH, DCN, ГРАФ, АЛГОРИТМ

**Haidur H.I. Use of Wireless MESH networks for increasing reliability management system in modern wireless technology.** In recent years wireless networks are widespread. MESH topology is used in information networks 3G, WiFi, WiMax, LTE. For such networks, the algorithm for calculating the optimal path aligned bandwidth, which can effectively improve the reliability MESH networks for modern wireless technologies.

**Keywords:** MOBILITY TECHNOLOGY, WIRELESS MESH, DCN, GRAPHS, ALGORITHM

За останні роки широке розповсюдження отримали безпроводові технології. І звичайно інтерес до цієї області не випадковий. Зручність використання незаперечні. Але побудова