

користувачу можливість у найкоротші терміни отримати всі необхідні ресурси. Ця можливість досягається завдяки таким властивостям cloud-систем як гнучкість, еластичність та віртуалізація. Разом з тим, велика кількість провайдерів cloud-послуг та відсутність чітких критеріїв порівняння останніх, ускладнює процес прийняття користувачем остаточного рішення щодо вибору cloud-системи. Також, одним із стримуючих факторів впровадження cloud-систем є відсутність методик оцінки їх надійності. Запропонована в роботі модель надійності cloud-систем, яка враховує особливості відомих типів відмов та взаємозв'язки між ними, дозволяє провести розрахунок та оцінити надійність надання сервісу відповідною cloud-системою. Саме рівень надійності надання послуг є визначальним для кінцевого користувача.

Подальші дослідження у даному напрямку доцільно зосередити на аналізі запропонованої моделі, доведенні її адекватності та апробації. Результатом цих досліджень повинна стати уточнена модель та метод визначення надійності cloud-систем.

### Література

1. Peter Mell The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology / Peter Mell, Timothy Grance. – NIST Special Publication 800-145. Computer Security Division. Information Technology Laboratory, Gaithersburg, MD 20899-8930. – 2011. – 7 p.
2. Foster The Grid 2: Blueprint for a New Computing Infrastructure / I. Foster, C. Kesselman. – Los Alamos, Morgan-Kaufmann, 2003. – 748 p.
3. C.S. Yeo Utility Computing on Global Grids. Technical Report / C.S. Yeo, R. Buyya, M.D. de Assunção, et al. – GRIDS-TR-2006-7, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia. – 2006. – 26 p.
4. Y. Zhang Transparent computing: A new paradigm for pervasive computing / Y. Zhang, Y. Zhou // Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC-06), LNCS 4145, 1– 11. – 2006. – P.18-34.
5. Bhaskar Prasad Rimal A Taxonomy and Survey of Cloud Computing Systems / Bhaskar Prasad Rimal, Eummi Choi, Ian Lumb // Proceedings of the Fifth International Joint Conference on INC, IMS and IDS. – 2009. – P.44-51.
6. Luis M. Vaquero A Break in the Clouds: Towards a Cloud Definition / Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, Maik Linder // ACM SIGCOMM Computer Communication Review. – 2009. – Volume 39. – P.50-55.
7. Mike Piech Platform-as-a-Servise Private Cloud with Oracle Fusion Middleware / Mike Piech. – An Oracle White Paper. – 2009. – 20 p.
8. Walter F. Witt Keep Your Feet on the Ground When Moving Software into the Cloud / Walter F. Witt // International Journal of Digital Content Technology and its Applications. – 2010. – Volume 4, Number 2. – P.10-17.
9. Локазюк В.М. Надійність, контроль, діагностика і модернізація ПК: Навчальний посібник для вузів / Локазюк В. М., Савченко Ю. Г. – Київ: “Академія” (Альма-матер), 2004. – 375с.

Надійшла 20.8.2011 р.

УДК 004.75

Д.М. МЕДЗАТИЙ, В.В. ДУДКО, Р.М. ХОМЕНКО

Хмельницький національний університет

## ОПТИМІЗАЦІЯ ВИДІЛЕННЯ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ У CLOUD-СИСТЕМАХ

*В роботі проведено класифікацію cloud-систем з метою визначення їх архітектурних особливостей. Проведено аналіз відомих методів вибору обчислювальних ресурсів для розв'язання конкретних задач. Запропоновано підхід до вибору обчислювальних ресурсів у cloud-системах за додатковим описом запиту, що дозволить підвищити ефективність опрацювання запитів та зменшить кількість відмов. Наведено приклад формування опису запиту для автоматизованої системи розподілу обчислювальних ресурсів.*

*In the article the classification and architecture specifics of cloud-systems have been considered. Analysis of known methods of computing resources choosing in cloud computing systems was conducted. The new approach for computing resources choosing, which use accessory description of user request was developed. New approach increase efficiency of request processing in the cloud systems.*

Ключові слова: обчислювальні ресурси, розподілені системи.

### Вступ

Сучасні комп'ютерні мережі дозволяють об'єднувати різнотипні обчислювальні засоби у єдину систему, здатну розв'язувати трудомісткі задачі, максимально ефективно використовуючи усі наявні вільні ресурси. Наявні високошвидкісні канали передачі даних дозволяють користувачам, незалежно від ступеня їх віддаленості отримати доступ до таких систем і користуватись необхідними ресурсами через мережу Інтернет.

В основі таких розподілених систем використовується ряд технологій, які дозволяють об'єднувати в єдину систему різні типи обчислювальні ресурси (кластери, Grid-системи); використовувати віртуалізацію на різних ієрархічних рівнях; змінювати конфігурацію системи під час роботи, з метою нарощування продуктивності; оцінювати об'єми виконаних робіт для оптимізації фінансових витрат користувачів. Одним з таких рішень є технологія cloud computing.

На сьогодні, не існує чіткого, стандартизованого визначення цієї технології. Національний інститут стандартів та технологій США (NIST) подає визначення cloud computing як технології (моделі обчислень), що дозволяє у прийнятній формі, за запитом отримати доступ до сумісно-використовуваного набору конфігурованих обчислювальних ресурсів (наприклад, мереж, серверів, систем збереження даних, програмного забезпечення, послуг тощо), які швидко можуть бути виділені та надані у користування при мінімальній взаємодії з провайдером послуг [1]. Cloud computing не є абсолютно новою технологією. Основні засади технології запозичені у grid computing (об'єднання різних обчислювальних ресурсів у єдиний кластер), utility computing (користувач сплачує лише за ті ресурси, які були йому надані і які він реально використовував), transparent computing (від користувача прихована вся складність внутрішньої архітектури, натомість йому наданий простий та ефективний інтерфейс для роботи з системою) [2-4]. Крім згаданих технологій, в cloud-системах використовуються і свої унікальні рішення, які вирізняють їх серед інших розподілених систем, наприклад, еластичність (обчислювальні ресурси, за запитом користувача, можуть оперативно нарощуватись не порушуючи цілісність та стабільність системи).

Разом з тим, cloud-системи мають і недоліки: відсутність чітких оцінок надійності; залежність швидкості відклику системи від багатьох факторів, що не дозволяє забезпечити високий рівень гарантоздатності; не завжди виділення обчислювальних ресурсів для конкретної задачі є оптимальним, та інші. Пошук підходів до вирішення перерахованих проблемних питань cloud-систем є актуальною науковою-практичною задачею, вирішення якої дозволить більш ефективно використовувати cloud-системи для вирішення специфічних науково-практичних задач.

### Постановка задачі

Надання послуг засобами cloud computing використовується у різних галузях, однак, абсолютна більшість – послуги щодо обслуговування Інтернет ресурсів. Це зручний підхід, який дозволяє власникам Інтернет ресурсів повністю абстрагуватися від технічних аспектів супроводження інфраструктури та зосередитися на якості програмного забезпечення, отримуючи від провайдера cloud-систем, необхідні ресурси за запитом. Разом з тим, технологія cloud computing отримує все більше прихильників і у інших, більш спеціалізованих галузях, наприклад, медицина [5], гена інженерія [6], біоінформатика [7] та ін. Використання технології cloud computing у специфічних галузях вимагає більш ефективних рішень, зокрема, у питанні виділення обчислювальних ресурсів для тієї чи іншої під задачі, яка буде вирішуватись. Загальні підходи не будуть ефективними, внаслідок відсутності можливостей щодо врахування особливостей конкретної задачі. Ці властивості не враховуються і через те, що провайдери послуг намагаються досягнути максимальної універсальності та швидкодії при виконанні типових задач. Тому для спеціалізованих cloud-систем актуальним є питання організації оптимального вибору обчислювальних ресурсів. Для вирішення цієї задачі в роботі необхідно: провести класифікацію cloud-систем для обмеження типів систем, на які можливо розповсюдити запропонований підхід; сформувані набір характеристик запитів, які визначають їх особливості та будуть враховуватись при виділенні ресурсів; описати набір властивостей обчислювальних систем, які дозволять адекватно оцінювати ефективність виконання отриманого запиту саме цими засобами.

### Класифікація cloud-систем

Класифікувати cloud-системи можна за декількома ознаками. Наприклад, у відповідності до того хто є поставником та користувачем послуг виділяють: приватні (private), публічні (public) та гібридні (hybrid) cloud-системи. При організації приватних cloud-систем провайдер та споживач послуг – це одна й та сама структура. Тобто, загальні принципи побудови системи зберігаються, але споживач одноосібно володіє всіма інформаційними ресурсами і сам керує доступом до них власних користувачів. Такий підхід дещо обмежує сферу застосування cloud-системи, але на порядок збільшує її захищеність. Публічні cloud-системи обслуговуються та налагоджуються провайдером cloud-послуг. Користувач через мережу Інтернет отримує доступ до замовлених ресурсів і користується ними. Гібридна система, як правило, об'єднує приватну та публічну cloud-систему, на які покладаються різні функції.

Більш інформативною є класифікація за видом сервісу, що надається (рис. 1).

В наведеній класифікації прийнято розрізняти: SaaS (Software-as-a-Service) – програмне забезпечення як сервіс; PaaS (Platform-as-a-Service) – платформа як сервіс; IaaS (Infrastructure-as-a-Service) – інфраструктура як сервіс. На рівні SaaS, користувач отримує у користування готове програмне забезпечення, може ним користуватись, завантажувати та зберігати власні дані, але не може керувати системними налаштуваннями, встановлювати та видаляти програмні засоби. На рівні PaaS, користувач отримує у користування платформу, в якій наявні засоби розроблення. Користувач може розробляти власні додатки, встановлювати, налаштовувати та видаляти їх, дозволяти доступ до власних додатків іншим користувачам. Разом з тим, не має доступу до апаратних засобів, обмежений можливостями встановленої та наданої йому у користування операційної системи та інших системних додатків. Рівень IaaS дозволяє користувачу власними

силами встановлювати та налаштовувати системні додатки, керувати віртуалізацією та розподілом ресурсів, але не має доступу до апаратного забезпечення. На всіх рівнях приховані від користувача ресурси (до яких він не має доступу) обслуговуються та підтримуються у робочому стані провайдером.

Подана класифікація може бути розширена, зокрема, у [8] запропоновано узагальнюючий підхід, в якому розглянуто модель ХааS, де X це програмне або апаратне забезпечення, інфраструктура, дані, платформа, бізнес і таке інше.

Повертаючись до спеціалізованих cloud-систем, слід зазначити, що їх реалізація передбачена або на рівні IaaS або у вигляді приватної системи. Організація приватної cloud-системи значно обмежує саму ідеологію cloud computing, тому не є доцільною. Отже, будемо розглядати задачу розподілу обчислювальних ресурсів для cloud-систем рівня IaaS.

**Архітектура підсистеми керування ресурсами cloud-систем**

Незважаючи на достатньо велику кількість рішень та провайдерів послуг cloud computing (Microsoft Azure, Amazon EC2, Google Apps Engine, IBM Blue Cloud, Nimbus, Oracle та ін.) єдиної архітектури не існує. Кожен з провайдерів при проектуванні закладає специфічні рішення, що вирізняють його серед інших. Однак, ряд складових cloud-систем є обов'язковим та присутнім у всіх рішеннях. Зокрема, таким компонентом є менеджер ресурсів (брокер ресурсів у Grid-системах). Основна задача менеджера – фіксувати наявність та поточний стан всіх ресурсів системи, а також, приймати кінцеве рішення про направлення поточного завдання на виконання до однієї із вільних підсистем. Оскільки основним рішенням при організації обчислювальних потужностей в cloud-системах є технологія grid computing, яка дозволяє об'єднувати в єдину систему різноманітні обчислювальні ресурси, то в cloud-системах зберігаються і основні принципи розподілу ресурсів притаманні grid-системам. Тоді, узагальнена архітектура підсистеми керування обчислювальними ресурсами cloud-систем може бути подана наступним чином (рис. 2).

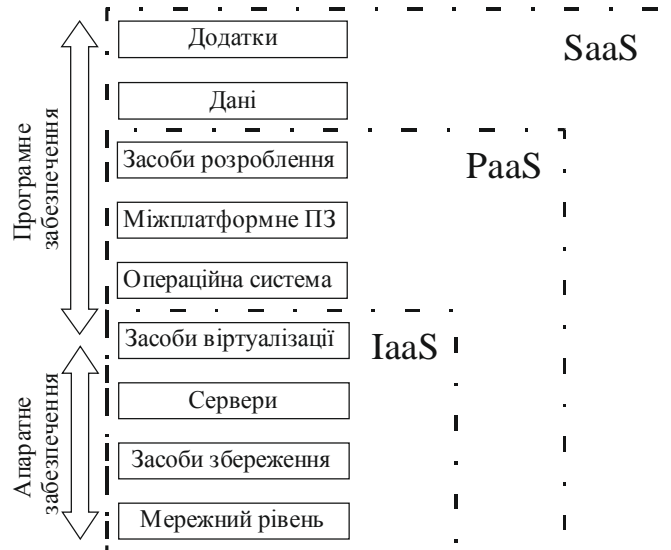


Рис. 1. Класифікація cloud-систем за ресурсами, що надаються

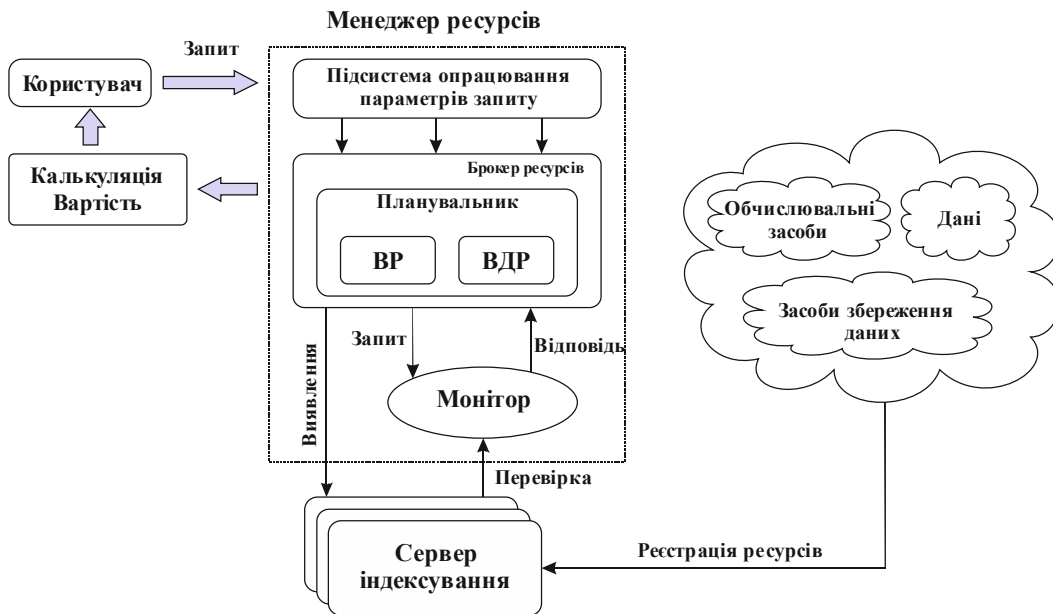


Рис. 2. Архітектура підсистеми вибору обчислювальних ресурсів

У поданій структурі логіка проходження запиту буде наступною. Користувач формує запит. Запит обробляється підсистемою опрацювання параметрів запиту. Опрацьовані дані потрапляють до брокера ресурсів. Брокер ресурсів містить два основних компоненти: ВР – компонент виявлення ресурсів (опрацювання інформації про наявні ресурси системи) та ВДР – компонент виділення ресурсів. Виявивши поточний стан ресурсів, брокер може при необхідності, актуалізувати для користувача вартість опрацювання даного запиту (під вартістю можуть матись на увазі – фінансові аспекти, час виконання, завантаженість ресурсів та таке інше). Компонент виявлення ресурсів отримує інформацію від серверу

індексування, який має актуальну інформацію про наявність ресурсів у cloud-системі. При виділенні ресурсів додатково проводиться актуалізація стану ресурсів засобами монітора ресурсів. У випадку якщо всі процедури виконані без відмов, запит користувача направляється на обрані обчислювальні ресурси для виконання. Результати опрацювання запиту повертаються користувачу.

Відзначимо, що наявність моніторингового компоненту, частково дозволяє уникнути відмов через відсутність виділених ресурсів. Інформація обов'язково актуалізується, від так, відмова через відсутність ресурсів може виникнути лише під час транспортування запиту чи даних до виділеного обчислювального ресурсу.

Варто зазначити, що такий підхід формування вибору обчислювальних ресурсів допустимий лише у спеціалізованих cloud-системах, в яких немає жорстких обмежень щодо оперативності опрацювання даних. У спеціалізованих системах, які використовуються для вирішення науково-практичних задач, що потребують специфічних та трудомістких обчислень. В таких системах оперативність обслуговування запитів не є вирішальною. Дослідники знаходяться перед вибором, опрацювати дані власними засобами, наприклад, тиждень, чи використати наявну cloud-систему та отримати результат через 10 годин. Зрозуміло, що за таких умов, додаткові кілька хвилин, що будуть витрачені на опрацювання запиту чи очікування потрібного ресурсу, є припустимими.

### Опис запиту

Для забезпечення можливості опрацювання та подальшої автоматизації процесу вибору обчислювального ресурсу, формалізуємо опис запиту користувача. Опис запиту може бути сформований користувачем або за допомогою автоматизованих програмних засобів. Зважаючи на той факт, що властивості запитів від одного і того ж користувача, як правило, співпадають (можливо не повністю, але у достатньо великій мірі), цей процес доцільно автоматизувати. Специфіка спеціалізованих cloud-систем дозволяє формувати запит у напівавтоматичному режимі, система у діалозі з користувачем уточнює та формує опис запиту. Нехай запит описується наступним набором даних:

$$M_{зан} = \langle D_{зан}, P_{зан}, T_{зан}, V_{зан}, H_{зан}, S_{зан} \rangle, \quad (1)$$

де  $M_{зан}$  - множина параметрів запиту;  $D_{зан}$  - підмножина параметрів опису вихідних даних;  $P_{зан}$  - підмножина параметрів оцінювання можливості розпаралелювання;  $T_{зан}$  - підмножина параметрів часової критичності;  $V_{зан}$  - підмножина параметрів, що описують очікувану точність;  $H_{зан}$  - підмножина параметрів, що описують вимоги до апаратного забезпечення;  $S_{зан}$  - підмножина параметрів, що описують вимоги до програмного забезпечення.

Оскільки мова йде про спеціалізовані cloud-системи, що розбудовують на рівні IaaS, то модель опису запиту може бути довільною, єдина умова, це узгодженість з менеджером ресурсів. Запропоновану модель слід розглядати як певний базовий варіант. Розглянемо більш детально кожен з підмножин.

Підмножина параметрів, що описують вихідні дані.  $D_{зан} = (d_1, d_2, \mathbf{K}, d_n)$ , де  $d_i, i = 1..n$  -  $i$ -й параметр, що описує вихідні дані (об'єм даних, тип даних, формат даних, інтенсивність звернення до даних та таке інше). Наприклад,  $d_1$  - об'єм даних до опрацювання (відбувається виділення необхідних ресурсів в залежності від об'єму даних заявлених користувачем, оцінюється навантаження на транспортування заданого об'єму даних);  $d_2$  - очікувана інтенсивність звернення до даних;  $d_3$  - необхідність шифрування (в разі необхідності шифрування даних враховуються необхідні додаткові ресурси);  $d_4$  - тип вихідних даних;  $d_5$  - формат подання даних та ін.

Відповідно до загальних засад розпаралелювання обчислювального процесу існують відомі методи оцінки можливості розпаралелювання за даними, за обчислювальним процесом. Разом з тим, оцінка можливості і необхідності розпаралелювання багато в чому залежить від апаратного забезпечення, на якому буде виконуватись обчислення. Крім того, важливим фактором при оцінці розпаралелювання є ефективність алгоритмів. Враховуючи те, що cloud-система розглядається як така, що зорієнтована в основному на представлення сервісу щодо обчислень, то передбачається, що алгоритм (кінцева програма) будуть надані користувачем. За таких умов, користувач здатний оцінити певні параметри, що будуть свідчити про можливість розпаралелювання. Тоді  $P_{зан} = (p_1, p_2, \mathbf{K}, p_m)$ , де  $p_j, j = 1..m$  -  $j$ -й параметр, що описує можливість розпаралелювання. Наприклад,  $p_1$  - складність обчислювального алгоритму (для однієї гілки, якщо наявне розпаралелювання);  $p_2$  - ступінь розпаралелювання (наявна у наданому алгоритмі);  $p_3$  - пов'язаність вихідних даних (ступінь можливості розпаралелювання за даними, чи необхідно буде організувати доступ до одних і тих самих даних для різних обчислювальних потоків, і чи будуть вони вносити зміни у вихідні дані).

Оскільки користувачу наперед не відомо, які саме апаратні засоби будуть використані, та зважаючи на об'єм вихідних даних, ступінь розпаралелювання, користувачу достатньо важко буде оцінити прогнозований час розв'язання задачі. Однак, користувачу відомий алгоритм, тому він може надати певні

попередні оцінки часу виконання (залежні від складності алгоритму) та очікуваний (бажаний) час отримання результатів.  $T_{зан} = (t_1, t_2, \mathbf{K}, t_l)$ , де  $t_k, k = 1..l$  -  $k$ -й часовий параметр. Наприклад,  $t_1$  - мінімально можливий час виконання паралельного алгоритму при використанні необмеженої кількості процесорів (визначається довжиною максимального шляху обчислювальної схеми алгоритму);  $t_2$  - час виконання програми при використанні одного процесора (час виконання послідовного варіанту алгоритму); для оцінки комунікаційної трудомісткості доцільно використати такий параметр як  $t_3$  - час початкової підготовки повідомлення (чи пошуку маршруту для передачі); допустимою є оцінка користувачем максимального часу очікування запиту, наприклад,  $t_4$  - час відклику сервісу (запит користувача видаляється із черги, якщо перевищено вказаний час).

Специфіка спеціалізованої cloud-системи буде визначати набори даних, які опрацюються. Однак, обслуговування інфраструктури, необхідної для експлуатації cloud-систем, є достатньо витратним. Тому, як правило, спеціалізовані cloud-системи мають певний рівень універсальності, який дозволяє варіювати в заданих межах. Це положення робить необхідним введення у структуру опису запиту набору параметрів  $V_{зан} = (v_1, v_2, \mathbf{K}, v_z)$ , де  $v_q, q = 1..z$  -  $q$ -й параметр, що описує необхідну точність обчислення. Параметри заданої підмножини містять таку інформацію як: точність вихідних даних, допустимі похибки обчислень, і таке інше.

За аналогією з рештою підмножин, формуються підмножини  $H_{зан}$  та  $S_{зан}$ , які регламентують наявність специфічних вимог до апаратного чи програмного забезпечення. Однак, в наслідок тієї чи іншої специфіки cloud-системи, ці вимоги можуть і не формуватися. Спеціалізована система повинна містити наперед визначений набір апаратних та програмних засобів. Разом з тим, ідеологія cloud computing дозволяє звертатись до сервісів великій кількості користувачів та розбудовувати систему на достатньо різномітному апаратному та програмному забезпеченні. Тому, недоцільно повністю відмовлятися від підмножин  $H_{зан}$  та  $S_{зан}$ . Але враховуючи їх специфіку, визначимо в цих підмножинах параметри, що будуть регламентувати необхідність опису. Наприклад,  $h_1 = 0 (s_1 = 0)$ , якщо додаткових вимог до апаратного (програмного) забезпечення немає, і, навпаки,  $h_1 = 1 (s_1 = 1)$ , якщо такі вимоги сформовано.

Відповідно до структури менеджера ресурсів (рис. 2), сформований запит від користувача подається до підсистеми опрацювання параметрів запиту, яка повинна визначити допустимість поданого опису, наявність відповідних ресурсів (зареєстрованих в системі) та сформулювати запит до брокера ресурсів cloud-системи.

### Опис обчислювальних ресурсів cloud-системи

Для автоматизованого прийняття рішення щодо виділення необхідних ресурсів, відповідно до параметрів запиту, на рівні підсистеми опрацювання параметрів запиту повинна міститись інформація про наявні в системі ресурси. Для оптимізації процесу опрацювання запитів, цю інформацію доцільно подавати у аналогічній формі. Тоді, будь-який ресурс системи описується множиною характеристик:

$$M_{рес} = \langle D_{рес}, P_{рес}, T_{рес}, V_{рес}, H_{рес}, S_{рес} \rangle, \quad (2)$$

де  $M_{рес}$  - множина параметрів, які характеризують ресурс;  $D_{рес}$  - підмножина параметрів, які визначають характеристики ресурсу за даними;  $P_{рес}$  - підмножина параметрів, що характеризують пристосованість ресурсу для опрацювання алгоритмів паралельної обробки даних;  $T_{рес}$  - підмножина параметрів, що характеризують часові аспекти обчислювального ресурсу;  $V_{рес}$  - підмножина параметрів, що характеризують допустиму обчислювальну точність даних засобів;  $H_{рес}$  - підмножина параметрів, що описують апаратне забезпечення ресурсу;  $S_{рес}$  - підмножина параметрів, що описують програмне забезпечення.

Кожна підмножина параметрів з  $M_{рес}$  повинна максимально відповідати відповідній підмножині з  $M_{зан}$ , але з певними особливостями. Наприклад, підмножина параметрів, що описують вихідні дані -  $D_{рес} = (d_1^p, d_2^p, \mathbf{K}, d_n^p)$ , де  $d_i^p, i = 1..n$  -  $i$ -й параметр, що характеризує специфіку обчислювального ресурсу щодо роботи з даними. А саме,  $d_1^p$  - максимальний об'єм вихідних даних, що можуть бути сприйняті (розміщені) обчислювальним ресурсом;  $d_2^p$  - максимально допустима інтенсивність звертань до даних;  $d_3^p$  - частка ресурсу, яка може бути виділена для додаткової (попередньої) обробки даних

(шифрування, архівація);  $d_4^p \mathbf{K} d_k^p$  – допустимі типи даних;  $d_{k+1}^p \mathbf{K} d_h^p$  – допустимі формати подання даних та ін.

Отримати однозначну відповідність між форматом подання запиту та описом обчислювальних ресурсів неможливо. Деякі параметри можуть мати пряму відповідність, інші ні. Крім того, при описі обчислювальної системи, намагаючись максимально наблизитись до опису запиту, доведеться частково дублювати дані. Наприклад,  $V_{pec}$  – частково буде містити інформацію подану у  $H_{pec}$ . Однак зважаючи на невеликі об'єми  $M_{pec}$  та  $M_{зан}$ , допустимо знехтувати цим дублюванням.

Враховуючи (1) та (2) задачу вибору обчислювального ресурсу можна сформулювати наступним чином. Нехай задано запит, що описаний множиною параметрів  $M_{зан}$ , та множину ресурсів системи  $M_{суст} = \{M_{pec}^i\}, i = 1..n$ , де  $n$  – кількість обчислювальних ресурсів системи. Необхідно з  $M_{суст}$  обрати таке  $M_{pec}^i$ , яке задовольняє  $M_{зан}$ . Для кожної підмножини, операція визначення відповідності буде мати свій сенс. Наприклад, підмножина  $D_{pec}^i = (d_1^p, d_2^p, \mathbf{K}, d_n^p)$  буде відповідати  $D_{зан}$ , якщо:  $d_1 \leq d_1^p$ ,  $d_2 \leq d_2^p$ ,  $d_3 \leq d_3^p$ ,  $d_4 \in \{d_4^p, \mathbf{K}, d_k^p\}$  та  $d_5 \in \{d_{k+1}^p, \mathbf{K}, d_n^p\}$ . Якщо всі умови дотримано, то  $i$ -й ресурс відповідає поточному запиту за вимогами до даних. Аналогічно визначаються вимоги  $i$  для інших підмножин.

### Висновки

Особливості технології cloud computing дають можливість організувати ефективні розподілені системи для вирішення спеціалізованих науково-практичних задач у різних галузях. Можливість побудови cloud-системи на рівні IaaS дозволяє реалізувати власні підходи до вирішення задачі вибору обчислювальних ресурсів.

Розглянута структура менеджера ресурсів обумовлює необхідність опису запиту користувача множиною параметрів. Запропонований опис обчислювальних ресурсів та структура запиту користувача дозволяє розробити автоматизований метод вибору обчислювальних ресурсів. Перевагою запропонованого підходу до опису параметрів запитів користувача та обчислювальних ресурсів є його універсальність, яка дозволяє розповсюдити його на широкий спектр спеціалізованих cloud-систем.

Для підвищення ефективності використання апаратних засобів cloud-систем необхідно розробити критерій відповідності обчислювальних ресурсів запиту. Критерій необхідний для прийняття остаточного рішення, якщо параметрам запиту відповідає декілька доступних обчислювальних ресурсів.

### Література

1. Peter Mell The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology / Peter Mell, Timothy Grance. – NIST Special Publication 800-145. Computer Security Division. Information Technology Laboratory, Gaithersburg, MD 20899-8930. – 2011. – 7 p.
2. Foster The Grid 2: Blueprint for a New Computing Infrastructure / I. Foster, C. Kesselman. - Los Alios, Morgan-Kaufmann, 2003. – 748 p.
3. C.S. Yeo Utility Computing on Global Grids. Technical Report / C.S. Yeo, R. Buyya1, M.D. de Assunção, et al. - GRIDS-TR-2006-7, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia. – 2006. – 26 p.
4. Y. Zhang Transparent computing: A new paradigm for pervasive computing / Y. Zhang, Y. Zhou // Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC-06), LNCS 4145, 1–11. – 2006. – P.18–34.
5. C. O. Rolim A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions / C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador // Second International Conference on eHealth, Telemedicine, and Social Medicine. – 2010. pp. 95–99.
6. Dennis P Wall Cloud computing for comparative genomics / Dennis P Wall, Parul Kudtarkar, Vincent A Fusaro, Rimma Pivovarov, Prasad Patil, Peter J Tonellato. – Bioinformatics. – 2010. – 12p.
7. Javier Bajo Cloud Computing for Bioinformatics / Javier Bajo, Carolina Zato, Fernando de la Prieta, Ana de Luis, Dante Tapia // Distrib. Computing and Artif. Intell., Springer-Verlag Berlin Heidelberg. – 2010. – p. 147–155.
8. Bhaskar Prasad Rimal A Taxonomy and Survey of Cloud Computing Systems / Bhaskar Prasad Rimal, Eummi Choi, Ian Lumb // Proceedings of the Fifth International Joint Conference on INC, IMS and IDS. – 2009. – P.44–51.

Надійшла 12.8.2011 р.