View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Technology audit and production reserves

ДОСЛІДЖЕННЯ СПОСОБУ ПОБУДОВИ ВЕКТОРНО-ПАРАМЕТРИЧНОГО БІСПЛАЙНА ЧЕТВЕРТОГО СТЕПЕНЯ З КЕРУЮЧИМИ ТОЧКАМИ, ЩО ЛЕЖАТЬ НА ПОВЕРХНІ

Досліджено спосіб побудови бісплайна (векторно-параметричної поверхні) за допомогою сплайна четвертого степеня з керуючими точками, що інцидентні поверхні. У результаті була отримана гладкість аж до третього порядку включно. Отримано алгоритм розрахунку бікубічної поверхні з першим, а потім другим і третім порядками гладкості. Наведено тестові приклади отриманих бісплайнів.

Ключові слова: векторно-параметричний сплайн четвертого степеня, бісплайн, сплайн з керуючими точками, що інцидентні кривій, гладкість.

Ковтун Александр Михайлович, *кандидат технических наук, доцент, кафедра общеинженерных дисциплин, Измаильский факультет Одесской национальной морской академии, Украина,* ***e-mail: ikra55@list.ru****.*

Ковтун Олександр Михайлович, *кандидат технічних наук, доцент, кафедра загальноінженерних дисциплін, Ізмаїльський факультет Одеської національної морської академії, Україна.*

Kovtun Alexander, *Izmail Faculty of Odessa National Maritime Academy, Ukraine,* ***e-mail: ikra55@list.ru***

УДК 519.687
DOI: 10.15587/2312-8372.2015.51415

Мацуєва К. А.

РОЗРОБКА МОДЕЛЕЙ І АЛГОРИТМІВ ОПТИМІЗАЦІЇ СПОЖИВАННЯ РЕСУРСІВ В СХОВИЩІ ДАНИХ НА БАЗІ ХМАРНОЇ ПЛАТФОРМИ

В рамках представленого дослідження побудована модель зберігання та організації розподіленого доступу до даних з використанням хмарної платформи мультимедійних ресурсів, розгорнутих в інформаційній системі. При цьому основним завданням дослідження є розробка алгоритмів і методів управління продуктивністю та оптимізація використання програмних і апаратних ресурсів.

Ключові слова: *розподіл навантаження, хмарні обчислення, СЗД, міграція даних, моделювання.*

1. Вступ

На сьогоднішній день однією з основних проблем при організації мультимедійних ресурсів є потреба в якісному наданні послуг кінцевим користувачам. Одним з найбільш розвинених напрямків, що використовують широкосмуговий доступ до мультимедійних послуг є організація досліджень і обчислень в сфері наукових досліджень. Ключовою особливістю застосовуваних технологій на відміну від традиційних мультимедійних сервісів є можливість надавати різні сервіси, використовуючи єдиний комплекс, що забезпечує інтерактивний зв'язок з користувачем за допомогою інформаційних каналів зв'язку. Це дозволяє застосовувати уніфіковані рішення в плані побудови архітектури таких сервісів. Крім того, відмітною особливістю послуг, що входять в таку інформаційну систему, є можливість прогнозувати поведінку користувачів. Це обумовлено специфікою процесу досліджень, а також регламентом роботи таких систем. Ці та інші чинники дозволяють здійснити вибір оптимального рішення, здатного забезпечити високу якість послуг, а також визначити основні механізми управління даними сервісами, враховуючи специфіку предметної області.

2. Аналіз літературних даних

Розглянемо основні алгоритми планування, що можуть бути використані для систем з архітектурою хмарних обчислень. Одним з найпростіших методів планування, що застосовуються для складання розкладу, є алгоритм First Come First Served (FCFS) [1, 2]. У кожному циклі планування, з черги виділяються запити і призначаються на визначені обчислювальні вузли. При цьому запити в черзі впорядковані згідно часу їх надходження. Крім описаного алгоритму застосовують і інші, що використовують список в якості основного елемента складання плану, включаючи Shortest Job First (найкоротша задача перша), Random Job First (випадкова задача перша) та ін. [2]. Істотний недолік спискових алгоритмів — низька завантаженість обчислювальних вузлів в силу наявності великої кількості вікон у створюваних розкладах, що призводить до простоювання та неефективного використання обчислювальних ресурсів.

Для вирішення цих проблем Аргонською національною лабораторією запропонований агресивний варіант алгоритму Backfill (алгоритму зворотного заповнення). Він переслідує дві конфліктуючі цілі — підвищення ефективності використання обчислювальних ресурсів шляхом заповнення порожніх вікон у розкладі та запобігання утримання заявок у черзі на обслуговування обчислювальним вузлом за рахунок механізму резервування. При цьому запити, що очікують, зберігаються в черзі і впорядковані відповідно до пріоритетів. У кожному циклі планування обчислювальні ресурси виділяються згідно встановленим пріоритетам. Особливістю алгоритму є те, що заявка, що надійшла на обслуговування, не

зможе отримати доступ до обчислювальних ресурсів, поки вони не відведені пріоритетним запитам. Таким чином, ресурси обчислювальної системи в першу чергу виділяються пріоритетним заявкам, а решта заповнюють вікна, що утворилися в ході резервування в розкладі в порядку, встановленому пріоритетами.

У дослідженнях Д. Фейтельсон і А. Вейл з Єврейського університету м. Єрусалиму запропонований консервативний варіант алгоритму Backfill, відмітною особливістю якого є більш жорстка залежність від пріоритетів. Запити, що надходять, не обслуговуються, поки в розкладі не виділені ресурси для більш пріоритетних заявок.

Дослідження показали, що консервативний варіант алгоритму Backfill по завантаженості обчислювальних вузлів не поступається агресивному варіанту, і, на відміну від останнього, дозволяє робити точні припущення про час запуску кожного завдання, що знаходиться в черзі [3]. Інша не менш важлива перевага консервативного варіанту алгоритму Backfill — можливість точного резервування ресурсів для запиту відразу ж після надходження в чергу. Це дає можливість застосовувати алгоритм Backfill в хмарних системах. Для роботи алгоритму Backfill суттєвою є наявність оцінок часу виконання запитів, що надходять в чергу. Однак при великій неточності в оцінках ефективність роботи алгоритму падає, а також знижується середня завантаженість обчислювальних вузлів.

Більшість широко відомих методів складання розкладів і вибору заявок з черги запитів не враховують топологію, використовувану в обчислювальній системі.

Найбільш перспективним напрямком на сьогоднішній день є використання технологій хмарних обчислень для побудови масштабованих систем, таких як системи організації досліджень. На ринку хмарних обчислень присутні не тільки пропрієтарні рішення, такі як VMware ESX, Xen та інші, але і добре документовані комплекси з відкритим вихідним кодом, такі як OpenStack [4].

В даний час існують рішення, побудовані на базі хмарних сервісів, що використовують універсальний підхід для організації доступу до розташовуваних в них ресурсів. При цьому не враховуються особливості кожного сервісу, що в свою чергу призводить до збільшення споживаних ресурсів і неефективного їх використання.

3. Постановка проблеми

У рамках дослідження встановлені наступні особливості споживання програмно-апаратних ресурсів, що використовуються для забезпечення роботи системи організації досліджень (СОД):

— навантаження на ключові ресурси носить періодичний і нерівномірний характер;

— одночасно відбуваються звернення до декількох типів ресурсів;

— інтенсивність звернення до кожного ресурсу може змінюватись в залежності від зовнішніх умов;

— зважаючи на відсутність розподілу навантаження між ресурсами при піковому навантаженні обладнання не завжди дозволяє обслужити всі запити;

— до 90 % навантаження заздалегідь відомо, оскільки для доступу до ресурсів використовується попередня реєстрація.

Крім того, варто відзначити, що 80 % ресурсів затребувані лише в 20 % часу роботи сервісів.

Встановлено, що єдиною точкою агрегації трафіку виступає система зберігання даних (СЗД), що забезпечує обробку потоку запитів, що надійшли від споживачів мультимедійних послуг. Отже, ефективність роботи всієї системи, а так само якість надаваних послуг безпосередньо залежить від продуктивності сховища даних. Тому для ефективного управління потоком запитів розроблена модель доступу до мультимедійних даних сховища хмарної системи.

Ключовою відмінністю сховищ мультимедійних даних є неоднорідність розміщуваної інформації (текстові, аудіо або відео дані) [5] і, як наслідок, різні підходи до організації доступу до неї. Крім методів доступу до даних істотною є інтенсивність звернення до тих чи інших елементів, яка може бути отримана з використанням внутрішньосистемних алгоритмів ідентифікації користувачів, що в свою чергу дозволяє оцінити затребуваність і спрогнозувати навантаження на пристрої системи зберігання. У зв'язку з цим важливим аспектом управління ресурсами системи, при значному збільшенні кількості одночасних запитів, є правильна організація процесу розміщення і розподілу елементів даних по пристроях [6, 7].

Відмінною характеристикою хмарних сховищ є реконфігурованість їх структури залежно від споживаних ресурсів. Це в свою чергу дозволяє впроваджувати алгоритми оптимізації в плані розміщення даних усередині дискового простору, а також керувати зміною кількості використовуваних системою пристроїв. При цьому процес оптимізації розміщення не повинен призводити до зниження якості обслуговування клієнтів СЗД, для чого в алгоритмах необхідно враховувати пропускну здатність мережі і максимальний обсяг даних, який можна передавати в один момент часу [8]. Крім того необхідно враховувати поточне завантаження самих пристроїв, а також їх розташування відносно один одного і клієнтів, що підключаються до них.

4. Об'єкт, мета та задачі дослідження

Об'єкт дослідження — сервіси інформаційної системи організації досліджень, що відносяться до декількох рівнів додатків і розміщені на базі хмарних систем.

Метою дослідження є визначення ключових параметрів, які впливають на роботу кожного з ресурсів, задіяних при побудові системи та оптимізація їх споживання з урахуванням розв'язуваної ними обчислювальної задачі.

Для досягнення поставленої мети необхідно розробити алгоритми і методи управління продуктивністю та оптимізацією використання програмних і апаратних ресурсів. Поставлено наступні задачі:

1. Побудувати модель обслуговування запитів користувачів в хмарному сховищі даних.
2. Розробити алгоритм балансування навантаження в хмарному сховищі даних.
3. Розробити алгоритм інтелектуальної міграції даних.

5. Результати досліджень сервісів інформаційної системи на базі хмарних обчислень

Для оптимізації механізмів доступу до даних побудуємо загальну модель доступу до даних системи зберігання.

Нехай:

$$R = (U, M, Q),$$

де $U = \{u_1, u_2, ...\}$ — множина користувачів; $M = \{m_1, m_2, ...\}$ — множина унікальних елементів даних, розташовуваних на пристроях зберігання. При цьому мінімальною одиницею даних m_i вважатимемо файл, що має обов'язкову властивість h — розмір.

Для забезпечення безпечного зберігання даних і балансування навантаження між пристроями зберігання визначимо функцію розподілу елементів даних, для цього введемо множину M_c:

$$M_c = \{m_1^{j_1}, m_1^{j_2}, m_1^{j_3}, ..., m_2^{j_1}, m_2^{j_2}, m_2^{j_3}, ...\},$$

де $m_i^{j_k}$ — k-я копія елемента розміщуваних даних (m_i) на j_k-м пристрої зберігання, за умови $k \geq 3$ (не менше трьох копій мінімальної одиниці зберігання на різних пристроях).

Тоді функція розподілу елементів даних по пристроях зберігання приймає вигляд:

$$P: M_c \to D.$$

Виходячи з викладеного вище, отримаємо вимогу користувача до елементів даних Q:

$$Q: U \to X \subseteq M_c,$$

де X — множина даних запитаних множиною користувачів U. Тоді сховище даних можна записати у вигляді кортежу:

$$S = (M_c, D, P, L, C, R, G),$$

де $D = \{d_1, d_2, ...\}$ — множина пристроїв зберігання; $L = \{l_1, l_2, ...\}$ — множина значень характеризує завантаження кожного пристрою зберігання (кількість одночасних звернень користувачів до конкретного пристрою); $C = \{c_1, c_2, ...\}$ — множина значень, що характеризує обсяг кожного з пристроїв у сховищі; $G \in N$ — натуральний коефіцієнт, що характеризує географічний (топологічний) пріоритет використання сховища.

Як правило, для великих хмарних структур використовуються консолідовані сховища, що складаються з ферм, і поєднують кілька сховищ в єдиний масив. Уявімо його як $S_{farm} = \{S_1, S_2, ...\}$.

Так як характеристики вимог користувачів змінюються в часі, перетворимо кортеж вимог $R(t) = (U, M_c, Q(t))$. Тоді $Q(t): U \to X \subseteq M_c$ — вимоги користувача до елементів даних, що змінюються в часі. Так як крім активності користувача змінюються властивості сховища, запишемо кортеж сховища в залежності від часу $S(t)$:

$$S(t) = (M_c(t), D(t), P(t), L(t), C, R(t), G),$$

де $D(t) = \{d_1, d_2, ...\}$ — множина пристроїв зберігання, змінних в часі, таких що $\forall t$, $D(t) > 0$; $P(t): M_c \to D$ — функція розподілу елементів даних між пристроями зберігання, змінних в часі.

При цьому для оптимізації витрат на апаратні ресурси і скорочення одночасно використовуваних пристроїв введемо кортеж відношень:

$$S_{cloud}(t) = \{S(t), D(t), D_{use}(t)\},$$

де $\forall t$, $D_{use}(t) \subseteq D(t)$ — множина пристроїв зберігання, що використовуються в масштабованому сховищі S в момент часу t.

Крім того, при масштабуванні сховища і міграції даних має виконуватись умова $\forall t, i, j \quad i \neq j \Rightarrow D_i(t) \cap D_j(t) = 0$, тобто при міграції даних сховища не повинні використовувати одні й ті ж пристрої. Це дозволить як гарантувати швидкість обробки інформації, так і забезпечити прийнятний час реконфігурації.

Таким чином, для мінімізації кількості одночасно використовуваних пристроїв зберігання в рамках одного масштабованого сховища і максимізації кількості оброблених запитів користувачів в одиницю часу, введемо цільову функцію виду:

$$\sum_{i=1}^{N} P_i(t) \to \min,$$

$$\sum_{i=1}^{N} L_i P_i(t) R_i(t) \to \max,$$

де N — загальна кількість заявок надійшли в систему на інтервалі часу ΔT.

На основі моделі доступу до даних сховища розроблено алгоритм балансування навантаження між пристроями, реалізований у вигляді програмного модуля для компонента Swift хмарної системи OpenStack. Вибір даної хмарної системи обумовлений відкритістю її архітектури та можливістю її модифікації під поставлені завдання. Основними недоліками OpenStack є неефективний алгоритм розподілу обчислювальних завдань між вузлами зберігання даних. Стандартний алгоритм, запропонований в системі, не враховує маршрутизацію віртуальної і топологію локальної мережі, а також віддаленість віртуальних машин, що виконують обробку запитів користувачів, і сховищ даних, що забезпечують передачу даних. Все це негативно впливає на час відгуку, як самої хмарної системи, так і запущених в ній примірників додатків. Крім того, самі алгоритми розподілу даних, застосовувані в сховищі хмарної системи, не дозволяють ефективно здійснювати розміщення інформації і надавати доступ до затребуваних даних по мережі [9–11].

Для оцінки ефективності розробленого алгоритму проведено моделювання роботи системи зберігання з різними параметрами. При цьому отримано такі закономірності при роботі стандартних алгоритмів хмарної системи.

При збільшенні кількості копій даних відбувається значне зниження навантаження на основних пристроях зберігання. Однак, при цьому зростає кількість задіяних пристроїв, що не відповідає поставленому завданню.

При одночасному доступі до декількох пристроїв, що містить різний обсяг даних, виникає дисбаланс

 ISSN 2226-3780

продуктивності сховища, що призводить до відмов в обслуговуванні запитів користувача.

Основною причиною є нерівномірне розміщення великих і малих за обсягом даних, що в свою чергу збільшує час зайнятості пристроїв.

При багаторазовому зверненні до тих самих даних в сховищі, пристрої не в змозі обслужити запити, оскільки відсутній розподіл навантаження між вузлами. При цьому застосовані в СЗД алгоритми кешування не можуть ефективно надати доступ до таких даних.

Розроблений алгоритм дозволяє врахувати перераховані недоліки роботи системи управління зберіганням даних, що в свою чергу, з урахуванням алгоритму пріоритетного обслуговування [10] дає додатковий приріст продуктивності хмари і розв'язуваних в ній завдань на 5–9 % у порівнянні зі стандартними засобами управління сховищем даних в OpenStack (рис. 1).

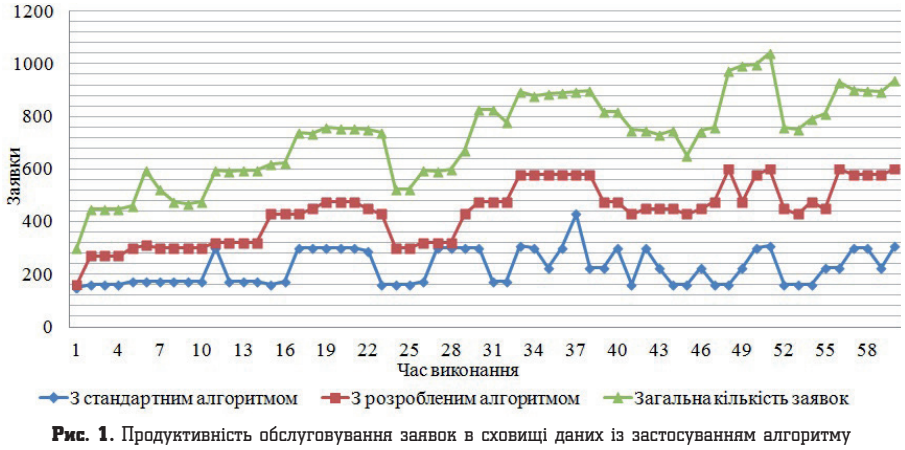

Рис. 1. Продуктивність обслуговування заявок в сховищі даних із застосуванням алгоритму інтелектуального кешування

Крім алгоритму розподілу навантаження важливим фактором, що впливає на продуктивність системи зберігання даних, є процес міграції даних між пристроями зберігання. Дана операція робить істотний вплив на час відгуку системи, так як дані в сховищі, як зазначалося раніше, є неоднорідними, а деякі з них є ще й залежними один від одного. Це особливо актуально при зверненні до потокових даних, наприклад при проведенні відео трансляції. Крім того, процес тиражування (процес розподілу даних між пристроями), у тому числі для кешування найбільш затребуваних даних так само безпосередньо залежить від ефективності алгоритмів, застосовуваних при міграції даних. Для оптимізації даного процесу, використовуючи можливості хмарної системи OpenStack, розроблений алгоритм, що формує план міграції даних, а також модуль, що здійснює розподілену обробку створених обчислювальних завдань.

При цьому всі операції, що задаються підсистемою планування можна описати як граф вимог G:

$$G(V,E,P),$$

де V — напрямок переміщення (кінцевий пристрій); E — елемент даних (файл), затребуваний на пристрої; P — пріоритет виконання операції в плані міграції.

У загальному вигляді схема взаємодії ресурсів у процесі міграції даних може бути представлена у вигляді наступної схеми (рис. 2).

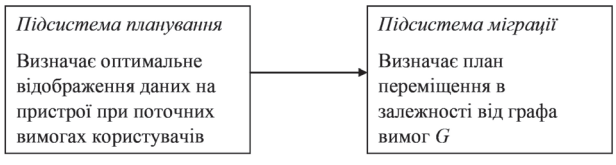

Рис. 2. Схема взаємодії підсистеми планування і підсистеми міграції

При формуванні плану міграції однією з основних особливостей є використання пріоритетного підходу при виборі операції.

Крім цього враховуються такі показники:

— поточна завантаженість вузлів;

— результати прогнозування навантажень, що спираються на історію звернень користувачів до тих чи інших елементів даних, а також на алгоритми внутрішньосистемної авторизації користувачів;

— розмір і тип затребуваних елементів даних;

— пропускна спроможність каналів зв'язку як зовнішніх, так і внутрішніх (в залежності від напрямку міграції даних);

— затребуваність активних даних, використовуваних в поточний момент (кількість користувачів звертаються до одного і того ж ресурсу в незалежності від його розташування в розподіленій системі зберігання).

Для складання обчислювальних завдань з міграції планувальником виділяються множини незалежних операцій DM_j. Вибір і об'єднання операцій в кожній множині визначається, по-перше, зв'язністю пристроїв, що беруть участь в поточній операції, по-друге зв'язністю напрямку міграції з іншими задачами. Кожній множині DM_j призначається пріоритет, рівний максимальному пріоритету операції, що входить в дану множину. Множини упорядковуються відповідно до розставлених пріоритетів. У ранжованому списку обчислювальних завдань виділимо два ключові множини і позначимо їх як DM_c і DM_{nc}. У множину DM_c віднесемо найбільш критичні операції в плані часу виконання, в DM_{nc} всі інші. Розроблений планувальник обчислювальних завдань спрямований на паралельну обробку двох підмножин. При цьому, на кожному етапі виконання обчислювальних задач проводиться аналіз зв'язків операцій кожного з множин, і складається оновлений ранжований список пріоритетів міграції, з урахуванням показників наведених раніше. Таким чином, розроблена система реального часу, що відслідковує стан пристроїв, розміщених на них даних, а також запити користувачів.

Провівши дослідну експлуатацію системи, із застосуванням алгоритмів кешування і інтелектуальність міграції даних, отримано сумарний приріст продуктивності хмари OpenStack на 15–19 % (рис. 3).

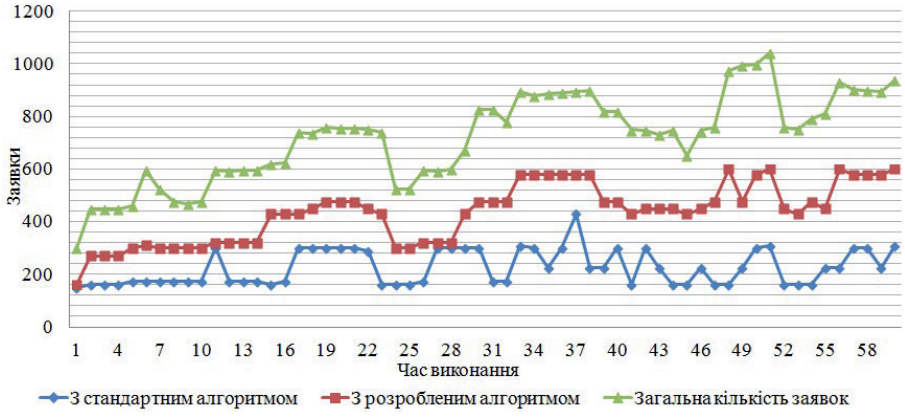

Рис. 3. Продуктивність обслуговування заявок в сховищі даних із застосуванням алгоритму пріоритетної міграції даних

6. Висновки

1. Розроблено модель доступу до мультимедійних даних сховища даних на базі хмарної платформи, що дозволяє мінімізувати кількість одночасно використовуваних пристроїв зберігання в рамках одного сховища і максимізувати кількість оброблених запитів за одиницю часу.

2. На основі моделі доступу до даних сховища розроблено алгоритм балансування навантаження між пристроями, що дозволяє знизити час відгуку, використовуючи інформацію про топологію і маршрутизацію основних потоків даних, а гнучке управління їх розміщенням дозволяє скоротити накладні витрати обчислювальних потужностей при міграції даних і віртуальних машин.

3. Розроблено алгоритм, особливістю якого є використання пріорітетного підходу при формуванні плану міграції і врахування більш широкого списку показників порівняно із стандартним алгоритмом системи OpenStack.

Комплексне моделювання роботи хмарної системи проводилося урахуванням особливостей компонентів мультимедійних ресурсів системи організації досліджень, при цьому використані інтелектуальні алгоритми дозволили масштабувати хмару, не знижуючи при цьому обсяги задіяних в роботі ресурсів. Розроблені модулі хмарної системи OpenStack показали свою ефективність в якості балансувальника навантаження, що дозволило надати ефективний доступ до користувачів різних типів даних.

Оцінка продуктивності показала зменшення часу обробки запитів за рахунок збільшення пропускної здатності системи при використанні розробленої технології.

Подальші дослідження будуть спрямовані на детальну розробку методики оцінки необхідного числа обчислювальних вузлів, необхідних для обслуговування запитів користувачів.

Література

1. Aida, K. Job Scheduling Scheme for Pure Space Sharing among Rigid Jobs [Text] / K. Aida, H. Kasahara, S. Narita // Lecture Notes In Computer Science, Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing. — London: Springer-Verlag, 1998. — P. 98–121. doi:10.1007/bfb0053983
2. Sinnen, O. Communication contention in task scheduling [Text] / O. Sinnen, L. A. Sousa // IEEE Transactions on Parallel and Distributed Systems. — 2005. — Vol. 16, № 6. — P. 503–515. doi:10.1109/tpds.2005.64
3. Гергель, В. П. Исследование алгоритмов планирования параллельных задач для кластерных вычислительных систем с помощью симулятора [Текст] / В. П. Гергель, П. Н. Полежаев // Вестник Нижегородского университета имени Н. И. Лобачевского. — 2010. — № 5(1). — С. 201–208.
4. OpenStack Open Source Cloud Computing Software [Electronic resource]. — Available at: \www/ URL: http://www.openstack.org/
5. Мацуєва, К. А. Методи управління ресурсами і додатками в обчислювальних системах на базі хмарних технологій [Текст] / К. А. Мацуєва // ScienceRise. — 2015. — № 7/2(12). — С. 33–38. doi:10.15587/2313-8416.2015.46591
6. Петров, Д. Л. Оптимальный алгоритм миграции данных в масштабируемых облачных хранилищах [Текст] / Д. Л. Петров // Управление большими системами. — 2010. — № 30. — С. 180–197.
7. Петров, Д. Л. Динамическая модель масштабируемого облачного хранилища данных хранилищах [Текст] / Д. Л. Петров // Известия ЛЭТИ. — 2010. — № 4. — С. 17–21.
8. Рогов, С. Тестирование производительности веб-серверов [Электронный ресурс] / С. Рогов, Д. Намиот // Открытые системы. — 2002. — № 12. — Режим доступа: \www/ URL: http://www.osp.ru/os/2002/12/055.htm
9. Ngenzi, A. Appling mathematical models in cloud computing: A survey [Text] / A. Ngenzi, R. Selvarani, Dr. Suchithrar // Jornal of Computer Engineering. — 2014. — Vol. 16, № 5. — P. 36–46. doi:10.9790/0661-16523646
10. Ruiz-Alvarez, A. A Model and Decision Procedure for Data Storage in Cloud Computing [Text] / A. Ruiz-Alvarez, M. Humphrey // Proceedings of the IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid'12). — Ottawa, 2012. — P. 572–579. doi:10.1109/ccgrid.2012.100
11. Мацуєва, К. А. Моделювання динамічного розподілення навантаження в інформаційній системі на базі хмарних обчислень [Текст] / К. А. Мацуєва // Вісник Національного технічного університету «ХПІ». — 2015. — № 22(1131). — С. 28–31.

РАЗРАБОТКА МОДЕЛЕЙ И АЛГОРИТМОВ ОПТИМИЗАЦИИ ПОТРЕБЛЕНИЯ РЕСУРСОВ В ХРАНИЛИЩЕ ДАННЫХ НА БАЗЕ ОБЛАЧНОЙ ПЛАТФОРМЫ

В рамках представленного исследования построена модель хранения и организации распределенного доступа к данным с использованием облачной платформы мультимедийных ресурсов, развернутых в информационной системе. При этом основной задачей исследования является разработка алгоритмов и методов управления производительностью и оптимизация использования программных и аппаратных ресурсов.

Ключевые слова: распределение нагрузки, облачные вычисления, СХД, миграция данных, моделирования.

Мацуєва Карина Андріївна, *аспірант, асистент, кафедра комп'ютеризованих систем управління, Національний авіаційний університет, Київ, Україна,* ***e-mail: karyna_matsueva@bigmir.net****.*

Мацуева Карина Андреевна, *аспирант, ассистент, кафедра компьютеризованных систем управления, Национальный авиационный университет, Киев, Украина.*

Matsueva Karyna, *National Aviation University, Kyiv, Ukraine,* ***e-mail: karyna_matsueva@bigmir.net***