Розвиток радіотехнічного забезпечення, АСУ та зв'язку Повітряних Сил

УДК 621.391

О.М. Місюра, В.Ф. Третяк, В.М. Більчук

Харківський національній університет Повітряних Сил імені Івана Кожедуба, Харків

МЕТОД ОПТИМІЗАЦІЇ СТРУКТУРИ РОЗПОДІЛЕНОЇ БАЗИ ДАНИХ У ВУЗЛАХ ІНФОКОМУНІКАЦІЙНОЇ МЕРЕЖІ ХМАРНОГО СЕРЕДОВИЩА

В статті розглядаються особливості побудови "хмарних" технологій, їх реалізація на програмному рівні. Показується, що сучасним підходом до проектування інформаційних систем є напрям хмарних обчислень (Cloud Computing), який містить спеціалізований спектр технологій обробки і передачі даних, коли комп'ютерні ресурси і потужності надаються як Інтернет-сервіси. Розглядаються механізми підтримки розподілених БД на основі фрагментації та реплікації даних.

Ключові слова: *інформаційні системи, "хмарні" сервіси, OLTP-система, база даних, Інтернет-сервіси.*

Вступ

Постановка проблеми в загальному вигляді. "Хмарні" технології є частиною нової мережевої інтернет-архітектури, яка базується на трьох основних принципах: інформаційно-орієнтованої мережевої архітектури (information-centric networking); "хмарних" обчисленнях, які інтегровані з мережею (cloud computing integrated with networking); відкритої конективності (open connectivity). "Хмари" відносяться до класу мережевих комп'ютерних систем, основними елементами яких є: комп'ютерна мережа з підвищеною надійністю і пропускною спроможністю; клієнт "хмари" – апаратне і програмне забезпечення, що взаємодіє з "хмарою" на основі стека протоколів TCP/IP; власне "хмара" – програмно-апаратний комплекс, що забезпечує роботові "хмарних" сервісів взаємодію з клієнтом і динамічне управління ресурсами хмарного середовища.

Особливостями "хмарних" технологій є наступні ознаки: сервісна модель обслуговування; самообслуговування; еластичність; використання поширених мережевих технологій. Зазвичай виділяють наступні базові класи "хмарних" сервісів: інфраструктура як послуга (Infrastructure as a Service, laaS); платформа як послуга (Platform as a Service, PaaS); дані як послуга (Data as a Service, DaaS); програмне забезпечення як послуга (Software as a Service, SaaS); робоче місце як послуга (Workplace as a Service, WaaS); усе як послуга (All as a Service, AaaS).

Сучасним підходом до проектування інформаційних систем є напрям хмарних обчислень (Cloud Computing), який містить спеціалізований спектр технологій обробки і передачі даних, коли комп'ютерні ресурси і потужності надаються як Інтернет-сервіси. Специфіка Cloud Computing полягає в тому, що забезпечується динамічне масштабування ресурсів хмари, його внутрішня структура прихована від споживача сервісів, використовується концепція плати у міру використання, пред'являються високі вимоги до надійності і доступності хмарної системи та ін.

Оскільки системи класу OLTP (On-Line Transaction Processing - системи оперативної обробки транзакцій), працюють з невеликими за розміром транзакціями, що поступають великим потоком, то клієнтові потрібний мінімальний час відгуку системи. Тому важливою вимогою стає обмін даними з OLTP-засобом – в реальному часі і з мінімальною затримкою. Ці показники безпосередньо залежать від використовуємих в OLTP-системах методів і архітектурних рішень, причому актуальним завданням є створення математичних моделей функціонування хмарної транзакційної системи, що дозволяють здійснювати імітаційне моделювання з метою отримання інтегральних показників ефективності її роботи.

У зв'язку з цим виникає завдання поліпшення отриманих характеристик, які можна вирішити різними способами (апаратні, програмні, архітектурні та ін.). Одним з варіантів підвищення продуктивності хмарної OLTP-системи являється оптимальне розміщення даних в хмарі. Таке завдання також виникає в умовах динамічного масштабування ресурсів хмари, коли при виході вузлів з ладу необхідно за мінімальний час визначити новий план розміщення даних і виконати їх міграцію з метою перерозподілу навантаження між іншими вузлами. Час отримання плану розміщення даних визначається складністю алгоритму, а час їх безпосередньої міграції залежить від характеристик технічних засобів і їх

завантаженості. Відповідно до типової угоди про рівень обслуговування (Service-Level Agreement, SLA), обидва етапи операції міграції мають бути виконані протягом 2–5 хвилин, тому час формування плану розміщення даних має бути мінімальним, а алгоритм рішення задачі розміщення даних в хмарі – мати високу швидкість.

Слід зазначити, що нині жодне велике підприємство не може обходитися без системи, що забезпечує функцій сховища даних. Все більше організацій прагнуть до активних операційних сховищ, тому оперативна обробка транзакцій є найважливішим засобом взаємодії з інформацією, що знаходиться в сховищах даних. Тим часом, побудова складних, високопродуктивних OLTP-систем є непростим завданням.

Аналіз досліджень і публікацій. Аналіз публікацій дозволив виділити наступні існуючі методи інтеграції додатків на рівні даних:

– консолідація даних. Технологія, що застосовується при такому методі, має назву ETL (Extract-Transform-Load, тобто Витягування-Перетворення-Завантаження). Цей метод призначений для вилучення необхідної інформації з різноманітних систем, перетворення між вихідним і цільовим форматом і завантаження в цільову систему (наприклад, в сховище даних). Основними недоліками цього методу є: затримка поновлення даних, оскільки дані копіюються з систем з певною періодичністю; підвищені вимоги до потужності цільового місця зберігання;

– федералізація даних. При такому методі кожне з n джерел містить n–1 фрагментів коду, що забезпечують трансляцію запитів до інших джерел федерації і перетворення результатів. Це забезпечує єдину віртуальну картину різнорідних джерел даних. Цей метод позбавляє від необхідності копіювати дані (наприклад, в сховище даних) і дозволяє використовувати дані безпосередньо з джерела. Основним недоліком цього методу є нелінійно зростаюча складність забезпечення віртуальної картини при збільшенні кількості джерел даних;

– поширення даних. За допомогою спеціальних програмних компонентів здійснюється копіювання даних між різними додатками. Копіювання може відбуватися в синхронному або асинхронному режимі. До основних недоліків цього методу можна віднести: підвищені вимоги до потужності споживача даних; обов'язкову присутність кожної програми в мережі при синхронному режимі, а при асинхронному режимі може виникнути ситуація, коли дані в додатках, що синхронізуються, не будуть співпадати;

– системи з медіатором. Медіатор - це програмний компонент, який забезпечує єдину точку входу для користувача запитів і єдине віртуальне бачення різнорідних джерел даних. Медіатор транслює запит користувача до джерел даних на основі загальної схеми і перетворює результати від джерел в єдину форму подання. При підключенні нового джерела даних потрібно створити відповідний адаптер. Основним недоліком такого методу є те, що дані доступні, як правило, тільки для читання;

– системи з посиланням на масив. При такому методі тиражуються в єдине місце зберігання не всі дані з кожного запису, а тільки частина, що використовуються для пошуку джерел даних, в яких містяться необхідні записи. До основних недоліків цього методу відноситься відсутність історичності даних і складна процедура емпіричного формування багатогранної структури єдиного довідкового масиву, зокрема, при додаванні нових джерел даних. Якщо контрольний масив оновлюється з деякою затримкою, то це негативно позначається на актуальності даних. Якщо ж оновлюється без затримки, то це може привести до нестачі ресурсів, необхідних для стабільного функціонування всієї системи (особливо при великій кількості джерел даних).

Виклад основного матеріалу

Розглянуті методи інтеграції даних є варіаціями двох основних механізмів підтримки розподілених БД:

– фрагментація даних - це розбиття БД або будь-якої її таблиці на фрагменти, які фізично зберігаються в різних БД, розташованих на різних вузлах комп'ютерної мережі і, можливо, управляються різними СУБД. Фрагментація даних дозволяє користувачам сприймати ці фрагменти так, як ніби вони працюють з локальною БД. Виділяють два основних види фрагментації таблиць: горизонтальна і вертикальна - це, відповідно, коли рядки і стовпці однієї логічної таблиці розподілені по декільком вузлам;

– реплікація даних - це процес копіювання даних з вихідної БД в цільову БД. При цьому дані можуть копіюватися інтенсивним або інертним способом. Інтенсивний спосіб передбачає, що зміни даних у вихідній БД будуть синхронно внесені в цільову БД як частина однієї транзакції. Інертний спосіб передбачає, що зміни даних з вихідної БД будуть асинхронно внесені в цільову БД в рамках вже іншою транзакцією. Практично перевага віддається інертному способу, щоб підвищити надійність роботи розподілених ІС, оскільки можна вносити зміни в вихідну БД без необхідності чекати внесення змін до цільової БД, але, оскільки зміни переносяться з певною затримкою, то в якийсь момент дані можуть відрізнятися.

Розглядається n – кількість вузлів мережі з довільною структурою; m – кількість незалежних фрагментів розподіленої бази даних (РБД); jj -й вузол мережі; F_iF_i - ii -й фрагмент РБД; L_iL_i -об'єм i -

го фрагмента; b_j – об'єм пам'яті вузла K_j, призначеного для розміщення фрагментів; s – кількість класів запитів (наприклад, читання, додавання, оновлення, видалення записів БД); λ_{ij}^k – інтенсивність запитів k-го класу $\left(k=\overline{1,s}\right)$ до фрагмента F_i ініційованих у вузлі; $\alpha_{ij}^k\alpha_{ij}^k$ – обсяг запиту k-го класу $\left(k=\overline{1,s}\right)$ до фрагмента F_i, ініційованого у вузлі; $\beta_{ij}^k\beta_{ij}^k$ – об'єм даних по запиту при виконанні запиту k-го класу $\left(k=\overline{1,s}\right)$ до фрагмента F_i, що поступив у вузол K_j.

Таким чином, об'єм даних, що пересилаються, при виконанні запиту k-го класу до фрагмента F_i, ініційованого у вузлі K_j, визначається таким чином: $\left(\alpha_{ij}^k+\beta_{ij}^k\right)\left(1-x_{ij}\right)$. При цьому $x_{ij}\left(i=\overline{1,m};j=\overline{1,n}\right)$ визначається наступним чином:

$$x_{ij}=\begin{cases}1, \text{якщо фрагмент F находиться у вузлі } K_j;\\ 0, \text{в інакшому випадку.}\end{cases} \quad (1)$$

Оскільки інтенсивність λ_{ij}^k породжує об'єм даних $\lambda_{ij}^k\left(\alpha_{ij}^k+\beta_{ij}^k\right)\left(1-x_{ij}\right)$, що потребують пересилки, то загальний об'єм даних, які необхідно переслати по каналам зв'язку між вузлами внаслідок функціонування розподіленої системи впродовж одиниці часу, визначається:

$$S=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{s}\lambda_{ij}^k\left(\alpha_{ij}^k+\beta_{ij}^k\right)\left(1-x_{ij}\right).$$

Якщо покласти, що $\lambda=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{s}\lambda_{ij}^k$, то цільова функція задачі оптимального розподілу фрагментів по вузлах ОМ буде мати вигляд:

$$V=\frac{1}{\lambda}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{s}\lambda_{ij}^k\left(\alpha_{ij}^k+\beta_{ij}^k\right)\left(1-x_{ij}\right).$$

Очевидно, чим менше значення середнього об'єму даних V, що пересилаються в одиницю часу, тим вище швидкість обслуговування запитів в системі.

Усі повідомлення, що поступають у вхідні черги вузлів, розподіляються на два типи: тип 1 - повідомлення, складові запити, для обробки яких необхідні фрагменти, які не зберігаються в БД вузла, і відповіді на ці запити; тип 2 - повідомлення, що становлять запити, для обслуговування яких потрібні фрагменти, які зберігаються в БД відповідного вузла. При цьому вважатимемо, що запит типу 1 для свого обслуговування у віддалений вузол перетворюється на запит типу 2.

Оскільки кожен фрагмент $F_i\left(i=\overline{1,m}\right)$ повинен знаходитися в одному з вузлів ОС, тоді

$$\sum_{j=1}^{n}x_{ij}\ge 1,\ i=\overline{1,m}.$$

Щоб наблизити модель до реальних систем, необхідно ввести коефіцієнт реплікації фрагментів RC. Цей параметр визначає кількість копій кожного фрагмента, розподілених по вузлах мережі. При цьому можливі два варіанти застосування цього коефіцієнта:

– коефіцієнт реплікації фрагментів RC визначає точну кількість копій кожного фрагмента (строга умова), тобто $\sum_{j=1}^{n}x_{ij}=RC,\ i=\overline{1,m}$;

– коефіцієнт реплікації фрагментів, який визначає максимальну кількість копій кожного фрагмента (нестрога умова), тобто

$$\sum_{j=1}^{n}x_{ij}\le RC,\ i=\overline{1,m}.$$

Тоді обмеження по кількості реплік фрагментів виглядатиме таким чином:

для строгої умови: $1\le\sum_{j=1}^{n}x_{ij}=RC,\ i=\overline{1,m}$,

для нестрогої умови: $1\le\sum_{j=1}^{n}x_{ij}\le RC,\ i=\overline{1,m}.$

Крім того, об'єм локальної БД кожного вузла $K_j\left(j=\overline{1,n}\right)$ не повинен перевищувати об'єм пам'яті цього вузла, призначений для розміщення фрагментів. Тому

$$\sum_{i=1}^{m}L_ix_{ij}\le b_j,\ j=\overline{1,n}.$$

Таким чином, завдання оптимального розподілу фрагментів по вузлах ОМ полягає в тому, щоб визначити значення змінних x_{ij}, де $\left(i=\overline{1,m};j=\overline{1,n}\right)\left(i=\overline{1,m};j=\overline{1,n}\right)$, які задовольняють умовам і дають мінімум лінійної функції. Отримана математична модель є задачею цілочисельного лінійного програмування з булевими змінними

Сутність запропонованого способу полягає у наступному. В блоці сортування даних по відношенню значення коефіцієнтів функціонала до різниці між максимальним та мінімальним значенням ваги матриці обмежень 1 виконується сортування:

$$\Psi_j=\frac{c_j}{\max\limits_i a-\min\limits_i a_{ij}},\quad \text{де}\quad a_{ij}=\frac{a_{ij}}{b_i} \quad (2)$$

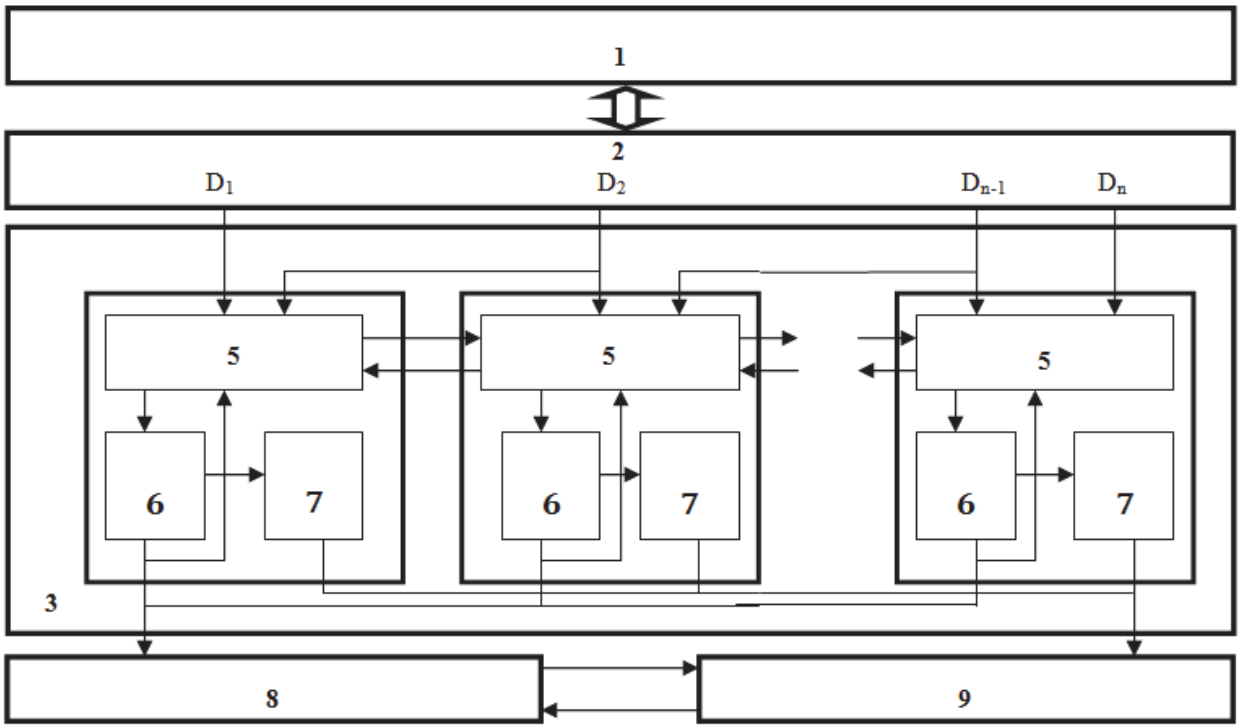

Рис. 1. Пристрій для рішення задач на графах

Обчислювальний пристрій 3 здійснює обчислення локальних екстремумів при заданому функціоналі та обмеженні, а також визначення (обчислення) номеру вершини, у якої локальний екстремум (ЛЕ) визначений за правилом

$$d_c\left(\mu_{sp}^{r}\right)+\gamma_p < \max_{\{c_j\}}\left\{d_c\left(\begin{matrix}*r\\ \mu_{sp}\end{matrix}\right)\right\}. \qquad (3)$$

З вершини *s* графа ΔD будується множина шляхів $m_{sj}^{r=1}$, $j=(\overline{1,n})$ першого рангу r, що задовольняє властивості, і в множинах $m_{sj}^{r=1}$ визначаються шляхи максимальної довжини $\left\{\begin{matrix}*r\\ \mu_{sj}\end{matrix}\right\}$ за вагою функціонала c_j. Для кожної вершини j визначається вага γ_j за правилом:

$$\gamma_j = c_{j+1}+c_{j+2}+\ldots+c_n,\quad \gamma_n=0;\quad j=(\overline{1,n-1}). \qquad (4)$$

Виключаються шляхи $\left\{\mu_{sp}^{r}\right\}$, $p=(\overline{r,n})$ у множині m_{sj}^{r} поточного рангу r, довжини якої $d_c\left(\mu_{sp}^{r}\right)$ задовольняють нерівності

$$\sum_{j=1}^{n} a_{ij}x_j \le b_1, \qquad (5)$$

$$x_j\in\{0,1\},\quad i=1,\ j=\left(\overline{1,n}\right),\ a_{1j}>0,\ c_j>0. \qquad (6)$$

Формується множина шляхів $m_{sp}^{r=r+1}$, $p=(\overline{1,n})$ наступного рангу, що задовольняє властивості, на базі множини шляхів m_{sj}^{r} попереднього рангу на основі правила відсікання неперспективних варіантів рішень по вибору мінімального значення довжини шляху в графі за вагою обмеження на основі принципу оптимізації за напрямком

$$p=\overline{r+1,n}\quad j=\overline{r,n}\quad j\ne p \quad p=\overline{r+1,n}\quad j=\overline{r,n}\quad j\ne p.$$

У визначених множинах $m_{sp}^{r=r+1}$ виділяються щонайдовші шляхи $\left\{\begin{matrix}*r=r+1\\ \mu_{sp}\end{matrix}\right\}$. Якщо визначиться декілька шляхів мінімальної довжини за вагою обмеження, то серед них вибирається шлях з найбільшим значенням довжини за вагою функціонала c_j.

Перевіряється, чи вся множина шляхів наступного (r+1)-го рангу порожня. Якщо умова виконується, то в множинах виділяється шлях максимальної довжини за вагою функціонала і алгоритм закінчує роботу. Якщо умова не виконується, то перевіряється $r=(n-1)$. У разі виконання рівності в множині виділяється шлях максимальної довжини за вагою функціонала і алгоритм закінчує роботу, інакше r збільшується на 1 і виконується обчислення.

Кожен процесорний елемент 4 обчислювального пристрою 3 виконує обчислення паралельно та здійснює обмін даними між сусідніми процесорними елементами після завершення обчислень. Блок регістрів 5 кожного процесорного елементу 4 зберігає і забезпечує мікрооперації передачі даних між регістрами блока регістрів сусідніх процесорних

елементів. Арифметичний обчислювач 6 обчислює локальні екстремуми на підставі даних, що надходять з блока регістрів, вибирає локальний екстремум за правилом (6) і пересилає його в обчислювальний пристрій формування вектора шляху 8 для обчислення глобального екстремуму та формування вектора шляху. Блок ідентифікації 7 визначає номер вершини, у якій локальний екстремум визначений на рис. 1.

Модуль пам'яті 9 зберігає номери вершин локальних екстремумів на кожному рангу обчислень. Дані $D_1, D_2, ... D_n$ надходять одночасно в кожну систолічну комірку обчислювального пристрою, в яких здійснюється обчислення. Введення даних здійснюється за допомогою блока управління систолічним процесом 2 із блока сортування даних по відношенню значення коефіцієнтів функціонала до різниці між максимальним та мінімальним значенням ваги матриці обмежень 1.

Список літератури

1. Третяк В.Ф. Метод оптимізації структури розподіленої бази даних у вузлах мережі хмарного середовища / В.Ф. Третяк, А.А. Корнієнко // Наука. Економіка. Інновації: Матеріали Міжнародної науково-практичної конференції, Чернівці, 15-16 січня 2017 р. - Т. 1. - Київ: Науково-видавничий центр «Лабораторія думки», 2017. - С. 7-9.

2. Патент на корисну модель № 9 2968, Україна, МПК G06 F15/00. Спосіб обробки та захисту інформації в розподілених сховищах даних / В.Ф. Третяк, В.В. Бараннік та ін. – № u201403994; заяв. 14.04.2014; опубл. 10.09.2014; Бюл. № 17. – 5 с.

Надійшла до редколегії 18.01.2017

Рецензент: д-р техн. наук В.В. Бараннік, Харківський національний університет Повітряних Сил імені Івана Кожедуба, Харків.

МЕТОД ОПТИМИЗАЦИИ СТРУКТУРЫ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ В УЗЛАХ ИНФОКОММУНИКАЦИОННОЙ СЕТИ ОБЛАЧНОЙ СРЕДЫ

О.М. Мисюра, В.Ф. Третяк, В.М. Бильчук

В статье рассматриваются особенности построения «облачных» технологий, их реализации на программном уровне. Показывается, что современным подходом к проектированию информационных систем является направление облачных вычислений (Cloud Computing), который содержит специализированный спектр технологий обработки и передачи данных, когда компьютерные ресурсы и мощности предоставляются как интернет-сервисы. Рассматриваются механизмы поддержки распределенных БД на основе фрагментации и репликации данных.

Ключевые слова: *информационные системы, "облачные" сервисы, OLTP-система, база данных, интернет-сервисы.*

METHOD OF OPTIMIZATION OF THE STRUCTURE OF THE DISTRIBUTED DATABASE IN THE NODES OF THE INFOCOMMUNICATION NETWORK OF THE CLOUD ENVIRONMENT

O.M. Misyura, V.F. Tretyak, V.M. Bilchuk

The article deals with the features of building "cloud" technologies, their implementation at the program level. It is shown that the modern approach to the design of information systems is the direction of cloud computing (Cloud Computing), which contains a specialized range of technologies for processing and transmitting data when computer resources and capacities are provided as Internet services. Mechanisms for supporting distributed databases based on data fragmentation and replication are considered.

Keywords: *information systems, "cloud" services, OLTP-system, database, Internet-services.*