Алгоритм міграції даних у масштабованому хмарному сховищі

Бур'ягін М.В.
НТУУ «КПІ», silentsnake@bigmir.net

In this work was described the concept of scalable cloud storage. A mathematical model and the modified greedy algorithm for coloring of the graph edges were proposed. Proposed algorithm's modification allows to reach more restrictions which are imposed on the model to approach the problem to a real.

ВСТУП

На сьогоднішній день хмарні технології та сервіси займають одну з провідних ролей в інформаційних технологіях та є найперспективнішими галузями на найближчі 5 років. Найбільш актуальними задачею є розробка масштабованих хмарних сховищ даних, які дозволяють динамічно змінювати структуру і кількість елементів збереження інформації.

МАТЕМАТИЧНА МОДЕЛЬ

Однією з задач, які вирішуються у масштабованих хмарних сховищах, є знаходження такого плану міграції, який має з мінімальну кількість кроків, що фактично означає мінімізацію часу міграції даних в масштабованому хмарному сховищі.

Модель масштабованого хмарного сховища можна зобразити за допомогою графу, вузли якого відповідають серверам, а зв'язки – фізичним зв'язком між серверами.

Основний граф розділяють на два підграфи: масштабуючий підграф G_S та підграф залишкової міграції G_R.

Під підграфом масштабування розуміють вузли (пристрої збереження інформації), які планують додати або видалити з хмарного сховища. Таким чином, граф можна представити у вигляді двох під графів (1):

$$G = < G_S, G_R > \quad (1)$$

Елементи підграфів G_S та G_R будемо позначати наступним чином:

a_{iS} - вершина підграфа G_S, де $i = 1...k$;

k - кількість вершин в графі G_S;

a_{jR} - вершина підграфа G_R, де $j = 1...n$;

n - кількість вершин в графі G_R;

$l(a_{iS}, a_{pS})$ – ребро, яке належить підграфу G_S та лежить між вершинами a_{iS} та a_{jS}, де $i \neq p, i = 1...k, p = 1...k$;

$l(a_{jS}, a_{mS})$ – ребро, яке належить підграфу G_R та лежить між вершинами a_{jS} та a_{mS}, де $j \neq m, j = 1...n, m = 1...n$;

$W(a_{iS})$ – коефіцієнт затримки обробки інформації вершини a_{iS} підграфу G_S;

$W(a_{jR})$ – коефіцієнт затримки обробки інформації вершини a_{jR} підграфу G_R;

c - кількість кольорів.

Введення в модель коефіцієнтів затримки обробки інформації на пристроях збереження дозволяє розширити задачу міграції та наблизити математичну модель до більш реального випадку. При побудові алгоритму міграції необхідно враховувати коефіцієнт затримки, і між

двома вершина обирати ту, в якої коефіцієнт затримки менший. Варто зауважити, що для коефіцієнта затримки справедливі обмеження (2) та (3):

$$W(a_{iS}) > 1 \quad (2)$$

$$W(a_{jR}) > 1 \quad (3)$$

Сумарний коефіцієнт затримки для двох вершин підграфу G_S (вершини (a_{iS}) та (a_{pS}) з'єднані одним ребром) можна записати так (4):

$$W(a_{iS}) + W(a_{pS}) \quad (4)$$

Деякі вершини можуть одночасно належати підграфу G_S та G_R, так як по ним відбувається роз'єднання та з'єднання підграфів G_S та G_R в єдиний граф G по якому і будується загальне розфарбування. Такі вершини будемо позначати і для якої виконується умова (5):

$$a_t, \text{ де } t = 1...q, a_{iS} = a_{jR} = a_t, \quad (5)$$

t - кількість спільних вершин.

Мінімальну кількість кроків для розфарбовування під графів G_S та G_R позначимо X_S та X_R відповідно. Тоді цільову функцію, в загальному, можна описати так (6):

$$\sum_{i=1}^{k} (W_{(a_{iS})}) X_S + \sum_{j=1}^{n} (W_{(a_{jR})}) X_R \to \min \quad (6)$$

Модифікація алгоритму

Задача міграції даних в масштабованих хмарних сховища є частковим випадком задачі міграції, та була сформульована Петровим Д. Л. у роботі [1] як задача багатокритеріальної оптимізації часу міграції. В загальному випадку, задача міграції даних є NP-складною, але її розв'язання може бути зведено до задачі розфарбування ребер графу та вирішено із застосування «жадібного» алгоритму.

«Жадібний» алгоритм – це раціональний евристичний алгоритм [2]. «Жадібним» його назвали із-за того, що ним намагаються зафарбувати якнайбільше вершин в один колір, при цьому уникаючи конфліктів. Евристичні алгоритми знаходять не оптимальний, а лише наближений розв'язок. Проте навіть наближений розв'язок сприймають як результат при розфарбуванні графу великих розмірів. Для застосування жадібного алгоритму введемо додаткові обмеження коефіцієнту затримки, та проведемо модифікацію класичного методу.

Крок 1. Обираємо ребро $l(a_{iS}, a_{pS})$ з G_S, яке лежить між вершинами з мінімальним сумарним коефіцієнтом затримки.

Крок 2. Призначаємо колір ребру.

Крок 3. Складається список ребер, які не зафарбовані. Для кожного з ребер визначається коефіцієнт затримки (4).

Крок 4. Обирається ребро, яке не виходить з тієї самої вершини та має найменший сумарний коефіцієнт затримки. Якщо таке ребро існує – йому присвоюється той самий колір, як і попередньому ребру.

Крок 5. Крок 4 повторюється доти, доки в списку не зафарбованих ребер є ребра, які не виходять з однієї вершини. Якщо в списку таких ребер немає, то береться наступне ребро і йому призначається наступний колір.

Крок 6. Алгоритм продовжує роботу, доки є можливість вибору ребер зі списку незафарбованих.

Далі алгоритм аналогічним чином застосовується для підграфу G_R. Після чого цього можна отримати загальний розв'язок.

Висновки

До переваг запропонованого алгоритму можна віднести можливість застосування до графів великої розмірності, можливість отримати кількість кольорів, достатніх для розфарбування графу.

До недоліків можна віднести те, що отримуємо лише наближений розв'язок (із-за евристичної природи алгоритму).

Література

[1] Петров Д.Л. Оптимальный алгоритм миграции данных в масштабируемых облачных хранилищах // Управление большими системами. - 2010. - Выпуск 30. – С. 180-197.

Асанов М.О. Дискретная математика: графы, матроиды, алгоритмы [Текст] / М.О. Асанов, В.А. Баранский, В.В Расин; СПб : Лань, 2010. – 368 с. – ISBN 978-5-8114-1068-2