УДК 681.324
ББК 32.973.202-018.2

ОПТИМАЛЬНЫЙ АЛГОРИТМ МИГРАЦИИ ДАННЫХ В МАСШТАБИРУЕМЫХ ОБЛАЧНЫХ ХРАНИЛИЩАХ [1]

Петров Д.Л. [2] , (Санкт-Петербургский Государственный Электротехнический Университет «ЛЭТИ», Сенкт-Петербург)

Исследуется многокритериальная задача оптимизации плана миграции данных в распределенном облачном хранилище. Разработан алгоритм миграции данных в масштабируемых облачном хранилище. Доказано, что алгоритм является полиномиальным и дает оптимальный результат по первому критерию.

Ключевые слова: облачные вычисления, распределенные хранилища данных, миграция данных.

Введение

Облачные вычисления сильно меняют взгляды на технологии [15, 3]. Главный аспект облачных вычислений это масштабирование [16]. Чем быстрее производится масштабирование тем больший эффект будет получен от использования облачных вычислений.

Особенно остро вопрос масштабирования стоит перед хранилищами данных [16]. Масштабирование хранилищ тесно

---

[1] Научный руководитлель - руководитель Центра Новых Информационных Технологий СПбГЭТУ, к.т.н., доцент Татаринов Ю.С.,

[2] Петров Дмитрий Леонидович, Начальник лаборатории (DLPetrov@mail.eltech.ru).

связанно с изменениями требований к данным и его целесообразно производить во время переконфигурации хранилища. Переконфигурация заключается в оптимизации распределения данных по устройствам хранения входящих в состав хранилища. Этот процесс требует перемещения больших объемов данных и занимает много времени.

Одновременное перемещение больших объемов данных приводит к резкому падению производительности. Поэтому обычно принимают, что любое из устройств хранения одновременно может участвовать не более чем в одной операции передачи данных. Также обычно принимают, что перемещаемые элементы данных имеют фиксированный размер и время передачи между любыми устройствами хранения. Задача составления оптимального плана перемещения данных в хранилище называется задачей миграции данных. Доказано, что эта задача NP-сложная [8], т.е. невозможно получить оптимальное решение за полиномиальное время. Существуют аппроксимационные алгоритмы ее решения [8, 7, 13].

Хранилища данных имеют фиксированное количество устройств хранения соединенных сетью. Облачными хранилищами будем называть хранилища основанные на инфраструктуре предоставляемой по требованию IaaS (Infrastructure as a Service). Облачные хранилища способны менять состав устройств хранения, т.е. способны масштабироваться. Устройства, добавляемые к хранилищу или высвобождаемые из хранилища, будем называть масштабируемыми устройствами. Добавление или удаление устройств должно происходить во время переконфигурации хранилища.

Задача миграции данных в общем виде является NP-сложной [8]. Задача миграции данных в масштабируемом облачном хранилище является частным случаем задачи миграции, она была сформулирована автором этой статьи в [14] как задача многокритериальной оптимизации плана миграции. Первый критерий задачи - оптимизация плана миграции на масштабируемых устройствах хранения, т.е. оптимизация времени масштабирования. Второй критерий - оптимизация плана миграции на остальных устройствах.

В этой статье представлен алгоритм решения задачи миграции данных в масштабируемых облачных хранилищах, а также доказывается полиномиальность алгоритма и его оптимальность по первому критерию.

1. Существующие алгоритмы

1.1. Задача распределения данных

Перед переконфигурацией хранилища необходимо вычислить оптимальное расположение данных в хранилище. Выходными данными этой задачи является отображение, в котором каждому элементу данных сопоставляются устройства хранения, на которых он должен быть после переконфигурации. Задача является NP-сложной, но были разработаны полиномиальные аппроксимационные алгоритмы ее решения [6, 12]. Вычислив новое расположения данных и имея старое расположение можно построить направленный мультиграф без циклов G демонстрирующий перемещение элементов данных из старой конфигурации в новую. Этот граф называют графом требований [8]:

$$\tag{1} \begin{cases} G = (V, E, P) \\ E \subseteq V \times V \\ P : E \to \mathbb{N} \\ \forall v, w \in V, P(v, w) = 0 \text{ если } (v, w) \notin E \end{cases}$$

V - устройства хранения, E - операция перемещения, P - весовая функция мультиграфа.

1.2. Задача миграции данных

План миграции можно разбить на шаги, поскольку все элементы данных имеют фиксированный размер и одинаковое время передачи. В этой работе мы не учитываем емкость устройств хранения. Задача миграции данных заключается в составлении плана перемещения данных между устройствами хранения согласно графу требований (1) за минимальное число шагов. Очевидно, что задача в таком виде элементарно сводится к задаче раскраске дуг мультиграфа. Минимальное число цветов раскраски дуг мультиграфа называют хроматическим индексом мультиграфа и обозначают χ'. Следует заметить, что направление передачи данных в хранилище не имеет значения с точки зрения задачи миграции [8]. Устройство считается занятым не зависимо от того, принимает оно данные или передает. Замена направленного мультиграфа на ненаправленный позволит сократить минимальное количество цветов раскраски:

$$\begin{cases} G = (V, E, P) \\ E \subseteq V \times V \\ P : E \to \mathbb{N} \\ \forall v, w \in V \Rightarrow P(v, w) = P(w, v) \end{cases} \tag{2}$$

1.3. Алгоритмы раскраски ребер мультиграфов

Задача раскраски дуг мультиграфа в общем виде является NP-сложной [10]. Существуют множество аппроксимационных, полиномиальных алгоритмов решения этой задачи [5, 9]. Для решения задачи будем использовать один из них:

Алгоритм 1 (Раскраска ребер мультиграфа). Описание полиномиального апроксимационного алгоритма раскраски можно найти в [9].

В работе не описываются существующие алгоритмы, а лишь приводятся ссылки на них.

Раскраска дуг двудольного мультиграфа является частным случаем задачи раскраски дуг, но эта задача не пренадлежит классу NP-сложных задач. Вероятно, первое решение этой задачи было предложено в работе [11]. Но существуют алгоритмы с меньшей трудоемкостью:

Алгоритм 2 (Раскраска ребер двудольного мультиграфа). Описание оптимального алгоритма раскраски можно найти в [2].

2. Миграция данных в масштабируемых облачных хранилищах

2.1. Модель масштабируемого облачного хранилища

Опишем модель масштабируемого облачного хранилища на основе хранилища без ограничений объема устройств хранения (2). Чтобы учесть особенности масштабируемого облачного хранилища явно выделим подмножество масштабирующих устройств хранения которые необходимо вывести из строя (или добавить) в результате переконфигурации.

Определение 1. Масштабирующим (scaling) подмножеством S будем называть подмножество вершин, которые необходимо вывести из строя или добавить к хранилищу в результате переконфигурации. Среди операций передачи данных не может быть операций передачи с одного масштабирующего устройства на другое, т.е.

$$\forall v, w \in S \subseteq V \Rightarrow P(v, w) = 0 \tag{3}$$

Определение 2. Масштабируемым облачным хранилищем G будем называть хранилище с явно выделенным масштабирующим подмножеством S.

$$\begin{cases} G = (V, E, P, S) \\ \forall v, w \in S \subseteq V \Rightarrow P(v, w) = 0 \end{cases} \tag{4}$$

2.2. Разбиение масштабируемого облачного хранилища

Во время масштабирования необходимо выполнить процедуру миграции на всех масштабирующих устройствах S. После этого их можно высвободить и выполнить миграцию данных на оставшихся устройствах хранения.

Разделим процедуру миграции на две части: масштабирование и остаточную миграцию. Для решения подобных задач часто используют методы разделения графов на подграфы [1, 4]. Разделим граф масштабируемого облачного хранилища G на два подграфа: масштабирующий подграф G_S и остаточный G_R.

Определение 3. Масштабирующим подграфом G_S будем называть подграф масштабируемого облачного хранилища G образованный множеством вершин S и множеством вершин S_{adj} смежными с S, а также всеми другими соединяющими вершины S с S_{adj}.

$$\begin{cases} G = (V, E, P, S) \\ G_S = (V_S, E_S, P) \\ V_S \subseteq V \\ E_S \subseteq E \\ v \in S_{adj} \subseteq V \\ \quad \Rightarrow \exists w \in S, P(v, w) > 0 \\ V_S = S \cup S_{adj} \\ (v,w) \in E_S \\ \quad \Rightarrow v \in S, w \in S_{adj}, P(v, w) > 0 \end{cases} \tag{5}$$

В последней формуле определения (5), выражение $v \in S, w \in S_{adj}, P(v, w) > 0$ эквивалентно выражению $v \in S_{adj}, w \in S, P(v, w) > 0$ поскольку граф G ненаправленный в соответствии с (2). На левой части рисунка 1 показан граф G, подмножество S (черные вершины) и S_{adj} (заштрихованные вершины).

Определение 4. Остаточным (residual) подграфом G_R будем называть подграф масштабируемого облачного хранилища G образованный всеми дугами графа G без масштабирующего подмножества S и все дуги, соединяющие эти вершины.

$$
(6) \qquad \begin{cases}
G = (V, E, P, S) \\
G_S = (V_S, E_S, P) \\
G_R = (V_R, E_R, P) \\
V_R \subseteq V \\
E_R \subseteq E \\
v \in S_{adj} \subseteq V \\
\quad \Rightarrow \exists w \in S, P(v, w) > 0 \\
V_R = V \setminus S \\
(v,w) \in E_R \\
\quad \Rightarrow v, w \in V_R, P(v, w) > 0
\end{cases}
$$

Очевидно, что множество вершин S_{adj} принадлежит как G_S так и G_R в соответствии с их определениями (5) и (6). Дуги, соединяющие вершины S_{adj} лежат только в G_R. На правой части рисунка 1 изображен граф, разделенный на две подграфа G_S и G_R.

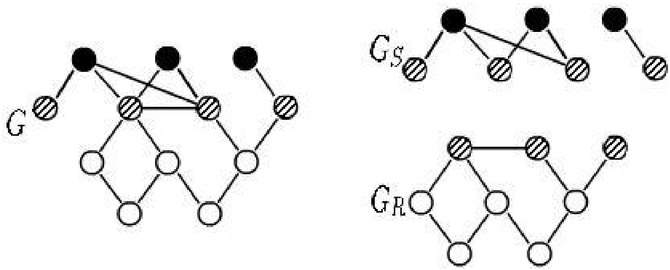

Рис. 1. Разбиение графа масштабируемого облачного хранилища G на два подграфа: G_S и G_R

2.3. Задача миграции данных в масштабирущем облачном хранилище

Используя введенные термины можно сформулировать задачу миграции данных в масштабируемом облачном хранилище.

Определение 5. Задача миграции данных в масштабируемом облачном хранилище G это многокритериальная задача оптимизации плана миграции. Основным критерием задачи является план миграции в подграфе G_S, а второй критерий - план миграции в подграфе G_R.

3. Алгоритм миграции данных в масштабируемом облачном хранилище

Опишем алгоритм миграции данных в масштабируемом облачном хранилище дающий оптимальный результат по первому критерию - оптимальность плана масштабирования. Для этого докажем, что масштабирующие подграф является двудольным мультиграфом для которых существуют оптимальные полиномиальные алгоритмы раскраски дуг.

Лемма 1. Масштабирующий подграф G_S является двудольным.

Доказательство. Исходя из определения масштабирующего подграфа (5), он состоит из множества вершин S и вершин S_{adj} смежных с S. Докажем что множества вершин S и S_{adj} разбивают граф G_S на две доли, т.е. в графе нет дуг идущих из S в S или же из S_{adj} в S_{adj}.

Вершины из множества S_{adj} не имеют общих дуг, поскольку это противоречит определению масштабирующего подмножества (3) входящего в состав масштабируемого облачного хранилища (4). Также из (5) следует, что любая его дуга соединяет вершины из S только с вершинами из S_{adj}. Следовательно, вершины из S_{adj} также не могут иметь общих дуг. Таким образом, множества вершин S и S_{adj} составляют две доли двудольного мультиграфа G_S.

Алгоритм 3 (Миграция данных). Опишем алгоритм в виде последовательности шагов описанных выше.

1. Выделить подграф G_S из графа G на основе известного подмножества S

2. Выделить подграф G_R из G на основе подграфа G_S

3. Использовать алгоритм 2 для расчета плана миграции двудольного мультиграфа G_S

4. Использовать алгоритм 1 для расчета плана миграции мультиграфа G_R

5. Получить общий плана миграции масштабируемого облачного хранилища G путем последовательного объединения план миграции мультиграфа G_S с планом миграции G_R.

Теорема 1 (Оптимальность алгоритма). Предложенный алгоритм 3 оптимален по первому критерию задачи миграции данных в масштабируемом облачном хранилище сформулированной в определении 5.

Доказательство. В соответствии с доказанной леммой 1, подграф G_S является двудольным мультиграфом и для нахождения оптимальной раскраски дуг G_S в алгоритме используется полиномиальный алгоритм 2, который дает оптимальный план миграции подграфа G_S. Исходя из постановки задачи сформулированной в определении 5 оптимальность алгоритма по первому критерию обеспечивается благодаря оптимальности плана миграции подграфа G_S. Т.е. алгоритм является оптимальным по первому критерию.

Теорема 2 (Полиномиальность алгоритма). Предложенный алгоритм 3 является полиномиальным.

Доказательство. Очевидно, что шаги 1 и 2 алгоритма 3 являются полиномиальными, поскольку вычисление вершин S_{adj} смежных с S - задача тривиальная, а выделение подграфов G_S и G_R, состоящих из заданных вершин, имеет не более чем квадратичную сложность $O(n^2)$. Полиномиальность шагов 3 и 4 доказана в [2] и [9] соответственно. Результат выполнения шагов 3 и 4 – два плана миграции, состоящие из последовательности операций перемещения данных (дуг подграфов), а шаг 5 состоит из простой операции последовательного объединения двух планов миграции. Очевидно, что шаг 5 имеет линейное время выполнения $O(k)$, зависящее от числа дуг k графа G .

Таким образом, алгоритм 3 является полиномиальным, поскольку он состоит из последовательности шагов без циклов, каждый из которых является полиномиальным алгоритмом.

4. Выводы

В статье описан алгоритм решения задачи миграции данных в масштабируемом облачном хранилище дающий оптимальный результат по первому критерию – скорости масштабирования. Оптимальная скорость масштабирования позволит облачным хранилищам предельно быстро арендовать и высвобождать устройства хранения данных и тем самым минимизировать срок и стоимость аренды устройств предоставляемых как инфраструктура по требованию IaaS.

Литература

1. BERGER M., BOKHARI S. Partitioning strategy for nonuniform problems on multiprocessors // IEEE Transactions on Computers. 1987. C-36(5). P. 570вЂ“580
2. COLE R., OST K., SCHIRRA S. Edge-coloring bipartite multigraphs in O(E log D) time // Combinatorica 21 (2001) 5-12.
3. DELIC K. A., WALKER M. A. Emergence of the academic computing clouds // ACM Ubiquity, Vol. 9, Iss. 31, 2008
4. GEORGE A., LIU J. Computer Solution of Large Sparse Positive Definite Systems // Prentice-Hall, Englewood Cliffs NJ, 1981
5. GOLDBERG M. K. Edge-coloring of multigraphs: Recoloring technique // J. Graph Theory, 8:121-137, 1984.

6. GOLUBCHIK L., KHANNA S., KHULLER S., THURIMELLA R. AND ZHU A. Approximation Algorithms for Data Placement on Parallel Disks // Proc. of ACM-SIAM SODA, 2000.
7. GOLUBCHIK L., KHULLER S., KIM Y. A., SHARGORODSKAYA S., WAN Y. Data Migration on Parallel Disks: Algorithms and Evaluation // Algorithmica 45(1): 137-158, 2006
8. HALL J., HARTLINE J., KARLIN A., SAIA J., WILKES J. On Algorithms for Efficient Data Migration // ACM Symposium on Discrete Algorithms, pp 620-629, 2001
9. HOCHBAUM D. S., NISHIZEKI T., SHMOYS D. B. A better than "Best Possible"algorithm to edge color multigraphs // J. off Algorithms, 7:79-104, 1996.
10. HOLYER I. The NP-completeness of Edge-Coloring // SIAM J. Comp., 11 (1982), 117-129.
11. HOPCROFT J., KARP R. An n. 5/2 algorithm for maximum matchings in bipartite graphs // SIAM J. Comput., 2 (1973), 225-231
12. KASHYAP S., KHULLER S. Algorithms for Non-Uniform Size Data Placement on Parallel Disks // Conference on Foundations of Software technology and Theoretical Computer Science, LNCS 2914, pp. 265вЂ“276, 2003.
13. KHULLER S., KIM Y.A. Algorithms for Data Migration with Cloning // SIAM Journal on Computing, vol. 33, No 2, pp 448 - 461, 2004.
14. PETROV D. L., TATARINOV Y. Data migration in the scalable storage cloud // IEEE, International Conference on Ultra Modern Telecommunications, ICUMT 2009, St. Petersburg
15. ROBISON S. A Bright Future in the Cloud // Financial Times, March 4, 2008
16. ZUCKERMAN B. Scalable storage in the cloud // Cloud Slam Conference, 2009, http://cloudslam09.com

OPTIMAL ALGORITHM FOR DATA MIGRATION IN SCALABLE STORAGE CLOUDS

Dmitry L. Petrov, Saint Petersburg Electrotechnical University «LETI», Saint Petersburg, Head of Laboratory (DLPetrov@mail.eltech.ru).

Abstract: The problem of multi-criteria optimization plan for data migration in distributed cloud storage is investigated. The algorithm of data migration in scalable cloud storage is developed. We proved that the algorithm is polynomial and gives the optimal result by the first criterion.

Keywords: computing clouds, distributed storage system, data migration.