

ПРИМЕНЕНИЕ ОДНОГО КЛАССА МАТРИЦ В ЗАДАЧЕ ВОССТАНОВЛЕНИЯ ДИСКА В СХД

А. Ю. Утешев^{1, *}, *Е. А. Калинина*^{1, **}

¹ Санкт-Петербургский государственный университет, Санкт-Петербург, Россия

Рассматривается проблема построения макета распределения фрагментов массива данных в системах хранения, использующих технологии декластерных RAID. Оптимальное в смысле равномерности распределения нагрузки при восстановлении фрагментов поврежденного диска решение проблемы задается посредством выбора циклических матриц с определенными свойствами. Устанавливается связь с задачей раскраски графа.

We treat the problem of data units distribution in data storage arrays based on the declustered Redundant Array of Independent Disks (RAID) technique. Optimal, with respect to balanced read-write load on disks in the process of reconstruction of the failed disk, solution to the problem is provided by the suitable circulant matrix. A relationship with the graph coloring problem is also established.

PACS: 07.05.Kf; 02.10.Ox

ОБЛАСТЬ ИССЛЕДОВАНИЯ

В сетевых системах хранения данных устройства могут быть организованы в одну или несколько групп дисков. Технология RAID использует одну или несколько таких групп для достижения более высокого уровня производительности и надежности. Когда RAID-система работает в условиях выхода из строя одного или нескольких дисков, скорость их восстановления становится решающей, особенно с учетом того, что любой дополнительный сбой диска может привести к безвозвратной потере данных. Это обстоятельство диктует необходимость задания такого распределения фрагментов данных по дискам, которое обеспечивает оптимизацию записи и близкий к идеальному параллелизм при реконструкции утерянных фрагментов. Различные подходы к решению задачи обсуждаются в [1, 2].

ОПТИМИЗАЦИЯ ВОССТАНОВЛЕНИЯ ПОВРЕЖДЕННЫХ ДИСКОВ

Структуру дискового массива с поддержкой конфигурации RAID5 проиллюстрируем таблицей.

* E-mail: a.uteshev@spbu.ru

** E-mail: e.kalinina@spbu.ru

Страйп	Диск 0	Диск 1	Диск 2	Диск 3	Диск 4	Диск 5	Диск 6	Диск 7	Диск 8	Диск 9
0	A_1	A_2	B_1	C_1	B_2	C_2	B_3	S	A_3	C_3
1	D_1	F_1	E_1	D_2	S	E_2	D_3	F_2	E_3	F_3
2	G_1	H_1	H_2	H_3	G_2	I_1	I_2	G_3	S	I_3
...	...									...

Массив содержит 10 физических дисков, на которых хранятся блоки данных (chunks), объединенные в группы по 3 (обозначены одинаковыми буквами). Каждая группа состоит из двух информационных блоков и одного служебного, содержание которого получено в результате XOR информационных. Каждую такую группу будем называть RAID-группой. Блоки каждой RAID-группы хранятся на разных дисках. В случае выхода из строя какого-то диска, содержимое его блоков восстанавливается путем вычисления XOR двух неповрежденных блоков затронутых RAID-групп. Результат сначала выкладывается не в поврежденный диск, а в специальные *запасные* блоки (spares) массива, обозначенные буквами S . И только после завершения этого процесса для всех блоков поврежденного диска производится перенос содержимого запасных блоков либо в старый, восстановленный диск, либо в новый свободный.

В общем случае будем обозначать G количество дисков в RAID-группе, а K — количество RAID-групп, которые можно разместить на различных дисках. Каждый такой набор из K групп будем называть *страйпом*. Для удобства страйпы удобно считать строками некоторой матрицы, а диски — ее столбцами. К каждому страйпу приписывается один запасной блок, расположенный в столбце, незадействованном в RAID-группах страйпа. Запись восстанавливаемого блока RAID-группы условимся производить в запасной блок того же страйпа. В матричной терминологии: процесс восстановления блока затрагивает исключительно только блоки той же строки. Наложим дополнительное условие: запасные блоки всех страйпов должны располагаться в различных столбцах (значение этого условия проясним ниже). Общее количество дисков считается равным $N := GK + 1$. Количество страйпов в массиве может быть произвольным, но в дальнейшем — для упрощения формулировок — будем считать его также равным N .

При выходе из строя диска 3 в приведенной схеме для восстановления блока C_1 задействованы диски 5, 7, 9 (два — для чтения и запасной — для записи результата XOR). Для восстановления блока D_2 задействованы диски 0, 4, 6, а для блока H_3 — диски 1, 2, 8. Заметим, что все задействованные диски имеют различные номера. Следовательно, восстановление всех трех утерянных блоков возможно производить одновременно, т.е. в параллельном режиме, без конфликтов доступа к дискам. Назовем любое такое объединение страйпов (строк) пучком, будем обозначать его номерами составляющих пучок строк: $\langle i_1, i_2, \dots \rangle$.

Состав пучка очевидно меняется при смене номера поврежденного диска. Так, при повреждении диска 0 строки 0 и 1 могут быть в составе одного пучка, в то время как строки 0 и 2 — нет (двойное обращение к диску 7).

Задача. Для любой комбинации чисел G и K составить такую $N \times N$ матрицу, чтобы при выходе из строя любого ее столбца кластеризация ее строк на непересекающиеся пучки была возможна в минимальном количестве последних.

Теорема 1. *Количество пучков в произвольной кластеризации строк матрицы на пучки не меньше, чем G .*

ЦИКЛИЧЕСКИЕ МАТРИЦЫ

Условимся обозначать блоки одной RAID-группы каждого страйпа одинаковыми числами от 1 до K ; при этом будем считать RAID-группы разных строк совершенно независимыми друг от друга, даже если их блоки обозначены одинаковыми числами. Таким образом, каждая строка искомой матрицы содержит ровно по G каждого из чисел $1, 2, \dots, K$ и вдобавок один запасной блок, который будем обозначать 0.

Поиск числовой матрицы, решающей поставленную задачу, будем производить в классе *циклических* матриц порядка N , т. е. матриц вида

$$\begin{pmatrix} b_0 & b_1 & b_2 & \dots & b_{N-1} \\ b_{N-1} & b_0 & b_1 & \dots & b_{N-2} \\ b_{N-2} & b_{N-1} & b_0 & \dots & b_{N-3} \\ \vdots & & \ddots & \ddots & \vdots \\ b_1 & b_2 & b_3 & \dots & b_0 \end{pmatrix} = [b_{(k-j) \pmod{N}}]_{j,k=0}^{N-1}.$$

Строки и столбцы матрицы нумеруем начиная с 0; нулевая строка матрицы называется *порождающей строкой*. Полагаем $b_0 := 0$.

Причину такого выбора кандидата на решение задачи поясняет следующий пример.

Пример 1. Для $G = 3$, $K = 4$ и циклической матрицы, порожденной строкой $(0, 1, 2, 3, 1, 3, 1, 4, 4, 2, 2, 4, 3)$, кластеризация строк на пучки в случае повреждения ее 0-го столбца возможна в виде

$$\langle 1, 2, 3, 7 \rangle, \langle 4, 5, 8, 9 \rangle, \langle 6, 10, 11, 12 \rangle$$

(0-ю строку не учитываем при кластеризации, поскольку содержимое запасного блока не требует восстановления). Оказывается, что кластеризация строк той же матрицы на пучки в случае повреждения ее 1-го столбца возможна в виде

$$\langle 2, 3, 4, 8 \rangle, \langle 5, 6, 9, 10 \rangle, \langle 7, 11, 12, 0 \rangle.$$

Вторая кластеризация получается из первой циклическим сдвигом номеров входящих в пучки строк.

Теорема 2. *Если при повреждении 0-го столбца циклической матрицы пучок $\langle i_1, i_2, \dots, i_\ell \rangle$ является допустимым для решения задачи кластеризации ее строк, то в случае повреждения h -столбца допустимым является пучок $\langle i_1 + h, i_2 + h, \dots, i_\ell + h \rangle \pmod{N}$.*

Ввиду этого результата решение задачи кластеризации конкретной циклической матрицы на пучки достаточно произвести для случая повреждения какого-то конкретного ее столбца; полученное решение тривиально адаптируется для случая повреждения любого другого столбца. Количество пучков и их мощности остаются инвариантными. И следующей ступенью в решении основной задачи становится задача оптимальной кластеризации в случае повреждения, для определенности, 0-го столбца матрицы. 0-ю строку при кластеризации не учитываем.

СВЯЗЬ С ЗАДАЧЕЙ ИЗ ТЕОРИИ ГРАФОВ

Чтобы соотнести задачу с конкретной областью математики, рассмотрим для каждой RAID-группы, содержащей блок из 0-го столбца, номера содержащих блоки группы столбцов. К этим номерам присоединим еще и номер строки, в которой содержится RAID-группа: он, очевидно, совпадает с номером столбца запасного диска той же строки. Полученный набор обозначим $\{0, j_1, \dots, j_G\}$ и назовем *шаблоном* RAID-группы. Это множество содержит номера всех дисков массива, которые будут задействованы для восстановления поврежденного блока. Каждому шаблону присваивается номер содержащей RAID-группу строки. Назовем эти строки *конфликтующими*, если их шаблоны содержат общие числа, отличные от 0. Очевидно, объединение в пучок допускают только строки, шаблоны которых не конфликтуют попарно.

Расставим номера строк $1, \dots, N - 1$ на плоскости и соединим отрезками пары номеров конфликтующих строк. В получившемся графе с $N - 1$ вершинами смежные вершины соответствуют конфликтующим строкам матрицы. Тогда задача о кластеризации строк матрицы на пучки превращается в задачу о раскраске вершин графа: когда требуется раскрасить каждую вершину графа так, чтобы смежные вершины были окрашены различными цветами [3]. Минимальное возможное количество цветов, необходимое для решения этой задачи, известно как хроматическое число графа, и в интерпретации нашей постановки задачи это число равно минимально возможному количеству пучков при кластеризации циклической матрицы, «порождающей» этот граф. В общей постановке задача вычисления хроматического числа графа является NP-сложной. Однако графы, порождаемые циклическими матрицами, имеют определенную специфику. Так, для любого из них хроматическое число гарантировано не меньше чем G (теорема 1). Авторами был разработан

алгоритм решения задачи, учитывающий эту специфику. Его тестирование на циклических матрицах небольшого порядка показало достаточную эффективность по сравнению с «жадными» алгоритмами, а также высокую вероятность успеха (в сравнении с абсолютно достоверными результатами, полученными в ходе полного перебора вариантов).

Решением задачи о раскраске конкретного графа проблема оптимального формирования дискового массива не ограничивается. Для заданных G и K требуется найти такую порождающую строку циклической матрицы, которой соответствует граф с минимально возможным хроматическим числом. Или хотя бы близким к теоретическому минимуму, а именно к G . Тотальный перебор всевозможных циклических матриц оказался осуществимым только для небольших значений G и K . Для $(G, K) \in \{(3, 4), (4, 4), (4, 5)\}$ теоретический минимум в G кластеров достижим. Так, для $G = 3$, $K = 4$ одна из таких оптимальных матриц приведена в примере 1. Эти матрицы достаточно редки. Так, для $G = 4$, $K = 4$ существуют только две матрицы с кластеризацией на 4 пучка. Для $G = 5$, $K = 4$ матриц с кластеризацией на 5 пучков не существует, а реально достигнутое, предположительно оптимальное, разбиение состоит из 7 пучков.

При более высоких значениях параметров тотальный перебор становится крайне ресурсозатратным («комбинаторный взрыв»). Достигнутые результаты в ходе частичного перебора упорядоченных определенным образом порождающих строк: 12 пучков для $G = K = 7$, 15 пучков для $G = K = 8$, 18 пучков для $G = K = 9$. На каждый из этих вариантов отводились сутки расчетов в следующей спецификации оборудования: CPU AMD EPYC processor 128 cores 7513, RAM 512, Drives 2 NVMe и 2 HDD SATA.

СПИСОК ЛИТЕРАТУРЫ

1. Wan J., Wang J., Xie C., Yang Q. S2-RAID: Parallel RAID Architecture for Fast Data Recovery // IEEE Trans. Parallel Distrib. Syst. 2014. V.25, No.6. P. 1638–1647.
2. Zhang G., Huang Z., Ma X., Yang S., Wang Z., Zheng W. RAID+: Deterministic and Balanced Data Distribution for Large Disk Enclosures // Proc. 16th USENIX Conf. on File and Storage Technologies. (FAST 2018). 2018. P. 279–293.
3. Jensen T.R., Toft B. Graph Coloring Problems. Wiley, 2011.