РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 835 373**(13) **C1**

(51) МПК
G06F 3/06 (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
G06F 3/0689 (2025.01)

(21)(22) Заявка: **2024118335, 02.07.2024**

(24) Дата начала отсчета срока действия патента:
02.07.2024

Дата регистрации:
25.02.2025

Приоритет(ы):
(22) Дата подачи заявки: **02.07.2024**

(45) Опубликовано: **25.02.2025** Бюл. № **6**

Адрес для переписки:
121205, Москва, ул. Луговая, 4, корп.2, Котлов Дмитрий Владимирович

(72) Автор(ы):
Васенина Анна Игоревна (RU),
Левицкий Иван Максимович (RU),
Смирнов Дмитрий Сергеевич (RU)

(73) Патентообладатель(и):
ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ШВАЧЕР" (RU)

(56) Список документов, цитированных в отчете о поиске: **RU 2747213 C1, 29.04.2021. RU 2502124 C1, 20.12.2013. US 9841908 B1, 12.12.2017. US 11163642 B2, 02.11.2021. US 10747617 B2, 18.08.2020.**

(54) СПОСОБ РАЗМЕЩЕНИЯ ДАННЫХ В RAID-МАССИВАХ ДЛЯ СБАЛАНСИРОВАННОГО РАСПРЕДЕЛЕНИЯ НАГРУЗКИ ВО ВРЕМЯ ВОССТАНОВЛЕНИЯ МАССИВА

(57) Реферат:

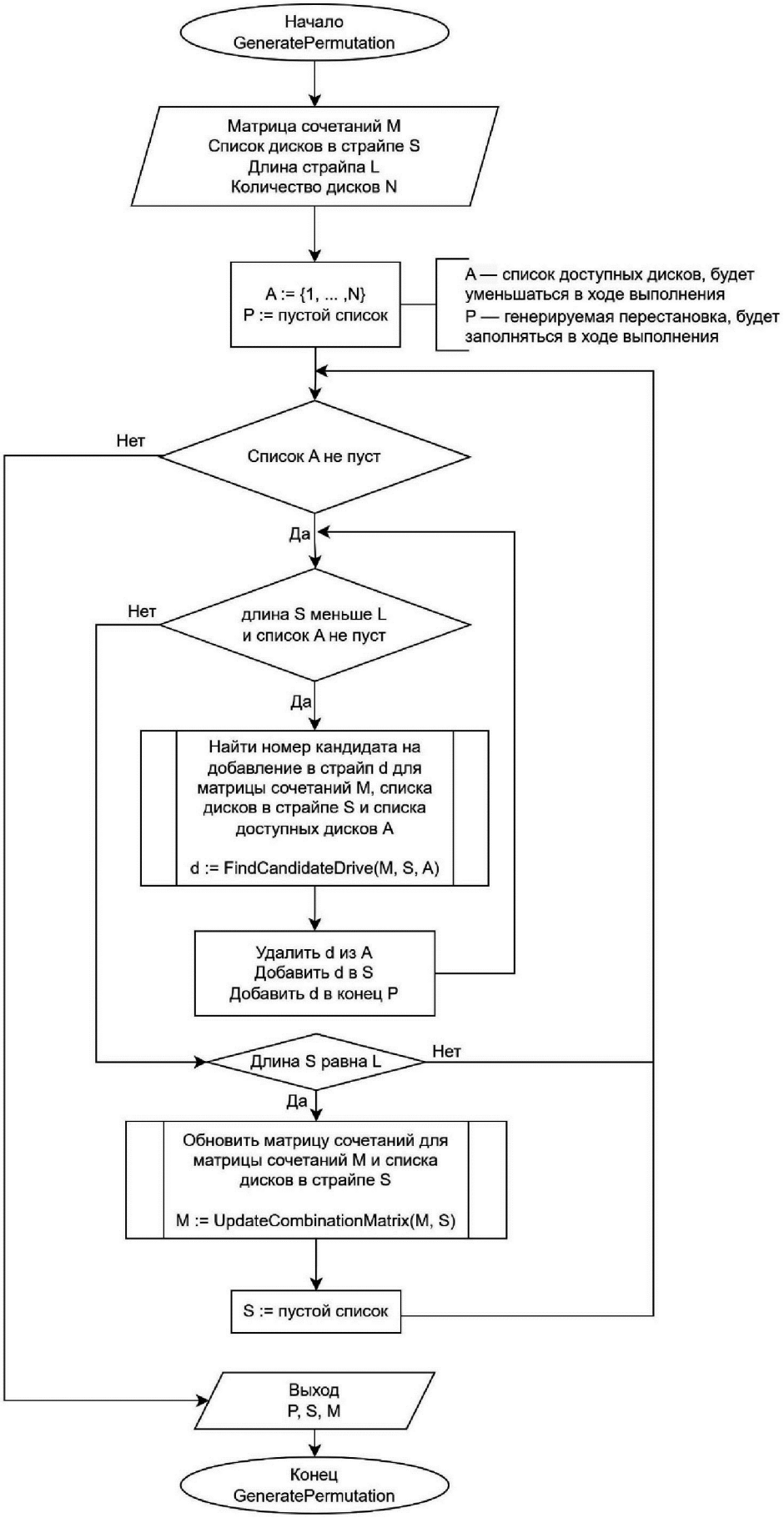
Изобретение относится к способу размещения данных в RAID-массивах для сбалансированного распределения нагрузки во время восстановления массива. Техническим результатом является увеличение скорости восстановления RAID-массива за счет схемы расположения данных, обеспечивающей равномерное распределение нагрузки чтения по всем дискам во время восстановления массива. Способ содержит этапы, на которых создают новый RAID-массив с помощью процедуры генерации карты размещения страйпов (generate stripe map); на вход процедуры генерации карты размещения страйпов передают количество свободных дисков N, длину текущего страйпа L и длину карты размещения страйпов R и на основе полученных данных формируют карту размещения страйпов, состоящую из R конкатенированных перестановок множества {1, …, N}; осуществляют инициализацию матрицы сочетаний M размером N×N, во время которой присваивают всем ее элементам значения 0, текущий страйп инициализируется пустым списком; выполняют генерацию перестановки в карте размещения страйпов путем вызова процедуры генерации перестановки (generate permutation) со следующими входными параметрами: текущее значение матрицы сочетаний M, список занятых дисков в текущем страйпе, длина текущего страйпа L, количество дисков в RAID-массиве N; для генерации одной перестановки на основе входных параметров осуществляют инициализацию вспомогательных структур, а именно: список свободных дисков инициализируется числами от 1 до N, список дисков в текущей перестановке инициализируется пустым списком; итеративно осуществляют выбор диска, который содержится в списке свободных дисков, но не содержится в списке занятых дисков в текущем страйпе, используя поиск минимальной суммы элементов матрицы сочетаний, соответствующих диску, и занятых дисков в текущем страйпе; добавляют диск в список занятых дисков в текущем страйпе и в конец списка дисков в текущей перестановке; как только длина списка занятых дисков в текущем страйпе достигла длины текущего

RU 2835373 C1

RU 2835373 C1

страйпа L, обновляют матрицу сочетаний и присваивают списку занятых дисков в текущем страйпе значение пустого списка; повторяют процедуру генерации перестановки в карте размещения страйпов до тех пор, пока количество перестановок не достигло R. 9 ил.

RU 2835373 C1



Фиг. 6

RU 2835373 C1

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 835 373**(13) **C1**

(51) Int. Cl.
G06F 3/06 (2006.01)

(12) ABSTRACT OF INVENTION

(52) CPC
G06F 3/0689 (2025.01)

(21)(22) Application: **2024118335, 02.07.2024**

(24) Effective date for property rights:
02.07.2024

Registration date:
25.02.2025

Priority:
(22) Date of filing: **02.07.2024**

(45) Date of publication: **25.02.2025** Bull. № **6**

Mail address:
121205, Moskva, ul. Lugovaya, 4, korp.2, Kotlov Dmitrij Vladimirovich

(72) Inventor(s):
Vasenina Anna Igorevna (RU),
Levitskij Ivan Maksimovich (RU),
Smirnov Dmitrij Sergeevich (RU)

(73) Proprietor(s):
OBSHCHESTVO S OGRANICHENNOJ OTVETSTVENNOSTYU "SHVACHER" (RU)

(54) METHOD OF ARRANGING DATA IN RAID ARRAYS FOR BALANCED LOAD DISTRIBUTION DURING ARRAY RECOVERY

(57) Abstract:

FIELD: physics.

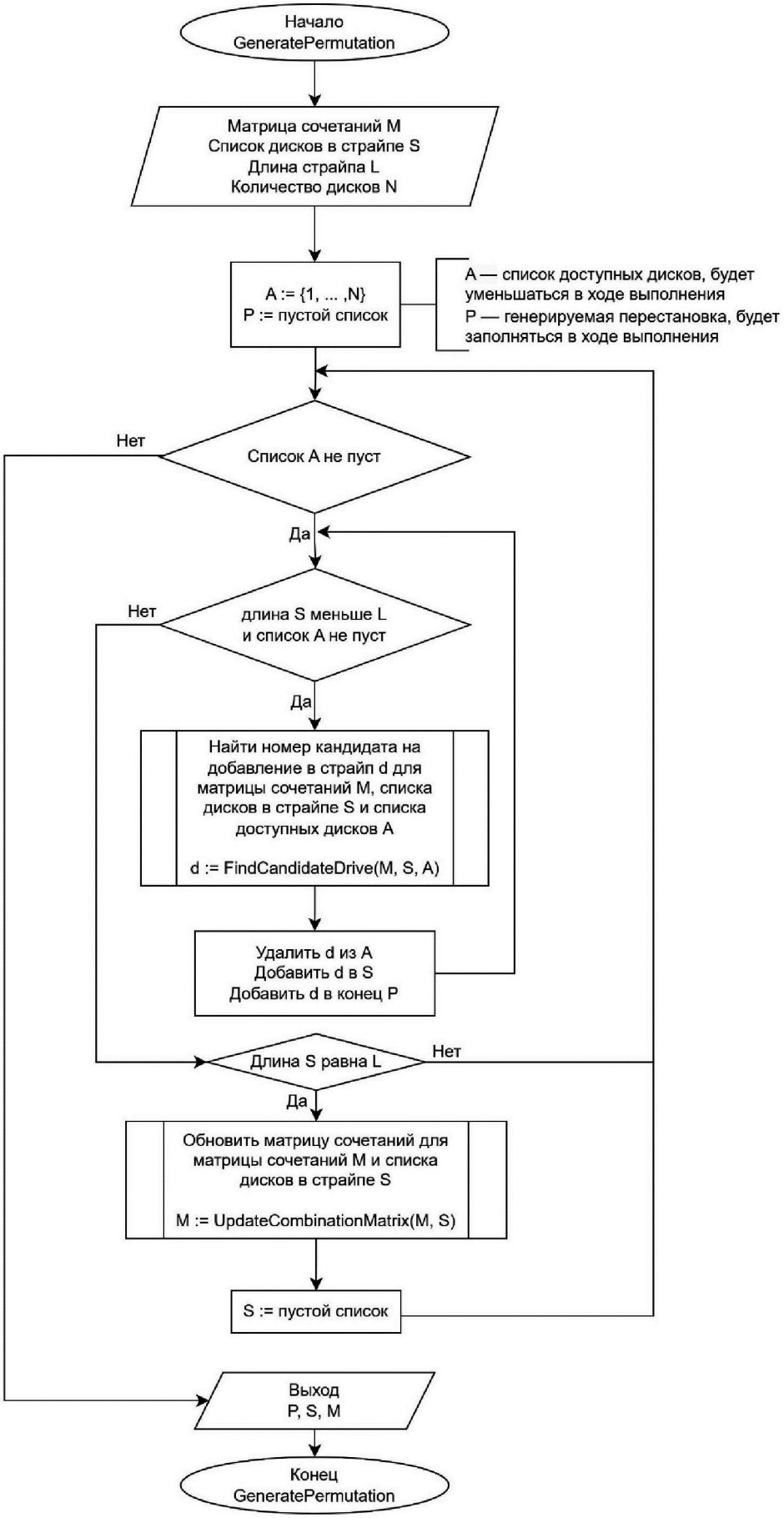
SUBSTANCE: invention relates to a method of arranging data in a RAID array for balanced load distribution during array recovery. Method comprises steps of creating a new RAID array using a procedure for generating a stripe map; number of free disks N, length of current stripe L and the length of the stripe arrangement map R is passed at the input of the procedure for generate a stripe map and based on the obtained data, a stripe map consisting of R concatenated permutations of the set {1, …, N} is formed; matrix of combinations M of size N×N is initialized, during which all its elements are assigned values of 0, the current stripe is initialized with an empty list; permutation is generated in the stripe map by calling the generate permutation procedure with the following input parameters: current value of combination matrix M, list of occupied disks in current stripe, length of current stripe L, number of disks in RAID array N; to generate one permutation based on input parameters, auxiliary structures are initialized, namely: list of free disks is initialized with numbers from 1 to N, list of disks in current permutation is initialized with empty list; iterative selection of a disk which is contained in the list of free disks, but is not contained in the list of occupied disks in the current stripe, using the search for the minimum sum of the elements of the combination matrix corresponding to the disk, and occupied disks in the current stripe; adding a disc to a list of occupied discs in the current stripe and to the end of the list of discs in the current permutation; once the length of the list of occupied disks in the current stripe has reached the length of the current stripe L, updating the combination matrix and assigning the list of occupied disks in the current stripe to an empty list value; procedure for generation of a permutation in a stripe arrangement map is repeated until the number of permutations reaches R.

EFFECT: faster recovery of a RAID array due to a data arrangement scheme which ensures uniform distribution of the read load across all disks during recovery of the array.

1 cl, 9 dwg

RU 2835373 C1

RU 2835373 C1



Фиг. 6

RU 2835373 C1

ОБЛАСТЬ ТЕХНИКИ

Заявленное техническое решение в общем относится к области вычислительной техники, а в частности к способу размещения данных в RAID-массивах для сбалансированного распределения нагрузки во время восстановления массива.

УРОВЕНЬ ТЕХНИКИ

Одной из важнейших характеристик систем хранения данных является доступность хранимых данных. Иными словами, возможность непрерывно работать с данными в течении длительного времени. При использовании стандартных устройств хранения данных, таких как жесткие диски (HDD) или твердотельные накопители (SSD) велика вероятность поломки устройства и потери доступа к данным. Одним из универсальных способов увеличить доступность данных, хранимых на устройствах, является объединение этих устройств в RAID-массив с избыточностью хранения данных. Избыточность данных в RAID-массиве может обеспечиваться хранением полных копий данных (RAID-1, RAID-10) или хранением контрольно-восстановительных сумм для некоторых фрагментов данных (RAID-5, RAID-6, RAID-50, RAID-60). Рассмотрим подробнее второй вариант.

RAID-массивы с использованием контрольно-восстановительных сумм разбивают все хранимые данные на последовательные фрагменты, называемые страйпами. Страйп состоит из стрипов данных равного размера. Для обеспечения избыточности к стрипам данных так же добавляют стрипы контрольно-восстановительных сумм (1 для RAID-5, 2 для RAID-6). Каждый стрип хранится на некотором физическом устройстве, при этом ни одно устройство не входит в один и тот же страйп дважды. Чем больше дисков в RAID-массиве, тем больше вероятность поломки одновременно нескольких дисков, в связи с этим увеличивают количество дисков с контрольно-восстановительными суммами. Для масштабирования RAID-6 на большое количество дисков используется технология RAID-60, в котором диски разделяются на несколько групп.Каждая группа образует подобие RAID-6, но данные при этом чередуются между группами. То есть, если есть 2 группы дисков, то нечетные страйпы будут лежать на первой группе дисков, а четные на второй.

Восстановление RAID-массива с контрольно-восстановительными суммами происходит по страйпам. Если по каким-то причинам оказывается невозможным прочитать стрип с данными, то вместо этого читается вся информация с других стрипов, в том числе со стрипов с контрольно-восстановительными суммами. Используя всю информацию, хранящуюся внутри страйпа и алгоритм восстановления с контрольно-восстановительных сумм, можно точно восстановить данные, пока число поврежденных (отсутствующих) стрипов не превышает количества стрипов с контрольно-восстановительными суммами.

При установке нового диска взамен поврежденного производится аналогичная процедура, но данные не только вычисляются, но и записываются на новый диск. Данный процесс называется процессом восстановления RAID-массива. В случае использования RAID-60 для восстановления дисков читаются только те диски, которые были в одной группе со сломавшимся диском, диски остальных групп не используются, что может приводить к длительной перегрузке дисков и новым поломкам.

Одним из решений этой проблемы являются альтернативные способы размещения данных, предложенные в заявленном решении.

Описанная логика RAID-массива может быть реализована двумя способами.

Первый способ - это использование аппаратного RAID-контроллера для управления массивом. В этом случае настройка производится до загрузки операционной системы,

и операционная система не видит базовых устройств хранения данных.

Второй способ - это написание драйвера устройства в ядре операционной системы, такие устройства называют программно-определяемыми или виртуальными. В этом случае диски видны ядру операционной системы, и уже после загрузки драйвера создается новое устройство, которое работает с переданными ему устройствами хранения данных, реализуя переадресацию запросов базовым устройствам согласно заданной логике.

Из уровня техники известен патент US9841908B1 «Declustered array of storage devices with chunk groups and support for multiple erasure schemes», патентообладатель Western Digital Technologies Inc, опубликован 12.12.2017. В данном решении описывается способ генерации сбалансированных неполных блок-дизайнов (balanced incomplete block designs, BIBD), способ генерации частичных сбалансированных неполных блок-дизайнов (PBIBD), а также способ применения сгенерированных блок-дизайнов для создания карты размещения страйпов (chunk group mapping table в терминологии патента). Генерация BIBD возможна только для конфигураций, описываемых формулой $N=k^2$, где N -количество дисков в RAID-массиве, k - количество дисков в страйпе. В ходе генерации используется случайно сгенерированная перестановка и последовательные операции вращения матриц (successive rotational operations). Генерация PBIBD используется только для тех конфигураций, для которых нельзя сгенерировать BIBD. Алгоритм использует последовательную псевдослучайную генерацию перестановок с оценкой параметров блок-дизайна на каждом шаге.

Недостатками описанного способа являются:

1. Применимость основного алгоритма генерации только для части возможных конфигураций.

2. Использование псевдослучайных генераций требует хранения дополнительной информации на дисках.

3. Время работы и длина вывода у алгоритма генерации PBIBD непредсказуемы и зависят от того, насколько удачно на каждой итерации генерируется перестановка.

4. Оцениваемые в патенте параметры блок-дизайна, такие как число блоков, содержащих любую точку, и число блоков, содержащих любые две точки, оценивают сгенерированный блок-дизайн только глобально, что может приводить к локальным перегруженным элементам.

Кроме того, из уровня техники известен патент EP2921960A2 «Method of, and apparatus for, accelerated data recovery in a storage system», патентообладатель Seagate Systems UK Ltd, опубликован 23.12.2015. В данном решении описывается способ раскладки данных, основанный на двух параметрах: ширине и количестве повторений, позволяющий оптимизировать восстановление RAID-массива через использование упреждающего чтения с базовых устройств. Подход состоит из трех основных шагов. На первом шаге определяются параметры ширины и количества повторений. Далее формируется матрица согласно выбранным параметрам, количеству дисков и количеству дисков в страйпе. На последнем шаге столбцы матрицы перемешиваются с помощью случайно сгенерированной перестановки.

Основными недостатками данного подхода являются:

1. При оптимизации за счет упреждающих чтений часть дисков оказывается перегруженной запросами, что может замедлять обработку пользовательских запросов и приводить к увеличению вероятности отказа дисков, с которых производится восстановление.

2. Использование псевдослучайной перестановки в конце генерации требует хранения

дополнительной информации на дисках.

К сожалению, жесткие диски, которые являются на сегодняшний день основным хранилищем данных, не так надежны, как хотелось бы. И достаточно остро стоит проблема обезопасить свои файлы, чтобы не пришлось прибегать к восстановлению данных.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Недостатки известного уровня техники преодолеваются и преимущества обеспечиваются посредством предоставления компьютерно-реализуемого способа размещения данных в RAID-массивах для сбалансированного распределения нагрузки во время восстановления массива.

Техническим результатом, достигающимся при решении данной проблемы, является обеспечение высокого уровня доступности данных, надежности хранения данных и увеличение скорости восстановления RAID-массива за счет схемы расположения данных, обеспечивающей равномерное распределение нагрузки чтения по всем дискам во время восстановления массива.

Кроме того, на разработанной схеме расположения данных удалось добиться максимального коэффициента несбалансированности не более чем 1.5 среди всех допустимых конфигураций (до 1024 дисков). За счет формирования страйпов с помощью матрицы сочетаний, хранящей в себе данные о том, сколько раз каждая пара дисков встречалась в одном страйпе осуществляют сбалансированное построение страйпов. Например, во время восстановления данных происходит чтение с тех дисков, которые входят в страйп с диском, восстановление которого производится.

К дополнительным эффектам можно отнести потребление оперативной памяти в объемах не более чем 2Мб (для худшего случая, 1024 диска) на один RAID-массив. Во время работы RAID-массива в оперативной памяти хранится только карта размещения страйпов. При фиксированной длине карты размещения страйпов в 1024 она хранит число элементов равное 1024 умножить на количество дисков. Для максимального количества дисков это 1024*1024 элемента. Размер элемента при этом 16 бит, так как этого достаточно для хранения чисел от 1 до 1024. Таким образом максимальное потребление памяти равно `1024*1024*16=16 777 216 бит=2 Мб.

Указанный технический результат достигается благодаря осуществлению способа размещения данных в RAID-массивах для сбалансированного распределения нагрузки во время восстановления массива, содержащий этапы, на которых:

- создают новый RAID-массив с помощью процедуры генерации карты размещения страйпов (generate stripe map);

- на вход процедуры генерации карты размещения страйпов передают количество свободных дисков N, длину текущего страйпа L и длину карты размещения страйпов R и на основе полученных данных формируют карту размещения страйпов, состоящую из R конкатенированных перестановок множества {1, …, N};

- осуществляют инициализацию матрицы сочетаний M размером NxN, во время которой присваивают всем ее элементам значения 0, текущий страйп инициализируется пустым списком;

- выполняют генерацию перестановки в карте размещения страйпов путем вызова процедуры генерации перестановки (generate permutation) со следующими входными параметрами: текущее значения матрицы сочетаний M, список занятых дисков в текущем страйпе, длина текущего страйпа L, количество дисков в RAID-массиве N;

- для генерации одной перестановки на основе входных параметров осуществляют инициализацию вспомогательных структур, а именно: список свободных дисков

инициализируется числами от 1 до N, список дисков в текущей перестановке инициализируется пустым списком;

- итеративно осуществляют выбор диска, который содержится в списке свободных дисков, но не содержится в списке занятых дисков в текущем страйпе, используя поиск минимальной суммы элементов матрицы сочетаний, соответствующих диску и занятых дисков в текущем страйпе;

- добавляют диск в список занятых дисков в текущем страйпе и в конец списка дисков в текущей перестановке;

- как только длина списка занятых дисков в текущем страйпе достигла длины текущего страйпа L, обновляют матрицу сочетаний и присваивают списку занятых дисков в текущем страйпе значение пустого списка;

- повторяют процедуру генерации перестановки в карте размещения страйпов до тех пор, пока количество перестановок не достигло R.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания и прилагаемых чертежей.

Фиг. 1 - иллюстрирует блок-схему выполнения заявленного способа;

Фиг. 2 - иллюстрирует состояния матрицы сочетаний после генерации трех перестановок в карте размещения страйпов;

Фиг. 3 - иллюстрирует требования к карте размещения страйпов;

Фиг. 4 - иллюстрирует пример итерации в генерации перестановки;

Фиг. 5 - иллюстрирует использование карты размещения страйпов для вычисления физического размещения страйпов;

Фиг. 6 - иллюстрирует блок-схему процедуры генерации перестановок;

Фиг. 7 - иллюстрирует блок-схему процедуры поиска кандидата на добавление в страйп;

Фиг. 8 - иллюстрирует блок-схему процедуры обновления матрицы сочетаний;

Фиг. 9 - иллюстрирует общий пример вычислительного устройства.

ДЕТАЛЬНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

В приведенном ниже подробном описании реализации изобретения приведены многочисленные детали реализации, призванные обеспечить отчетливое понимание настоящего изобретения. Однако, квалифицированному в предметной области специалисту, будет очевидно каким образом можно использовать настоящее изобретение, как с данными деталями реализации, так и без них. В других случаях хорошо известные методы, процедуры и компоненты не были описаны подробно, чтобы не затруднять излишне понимание особенностей настоящего изобретения.

Кроме того, из приведенного изложения будет ясно, что изобретение не ограничивается приведенной реализацией. Многочисленные возможные модификации, изменения, вариации и замены, сохраняющие суть и форму настоящего изобретения, будут очевидными для квалифицированных в предметной области специалистов.

Ниже будут описаны понятия и термины, необходимые для понимания данного технического решения.

RAID или RAID массив (англ. Redundant Array of Independent Disks - избыточный массив независимых (самостоятельных) дисков) - совокупность из нескольких блочных энергонезависимых устройств хранения, например дисков (SSD, HDD), объединенных в единое логическое блочное устройство таким образом, что выход из строя одного или нескольких блочных устройств в составе RAID не вызывает выхода из строя самого массива, и не приводит к потере данных.

Стрип (Strip) - последовательный участок базового устройства хранения данных (диска RAID-массива) фиксированного размера, например 16 килобайт.

Страйп (Stripe) - набор стрипов, расположенных на разных базовых устройствах хранения (дисках RAID-массива) и вместе формирующих последовательный участок виртуального устройства хранения (RAID-массива). Каждый страйп содержит набор данных, а также, опционально, контрольно-восстановительные суммы, вычисляемые от набора данных страйпа. В случае хранения контрольно-восстановительных сумм, под них выделяются отдельные стрипы (по количеству различных контрольно-восстановительных сумм). Глубиной страйпа (Stripe depth) называется размер одного стрипа, входящего в состав страйпа. Шириной страйпа (Stripe width) называется объем данных, содержащийся в каждом страйпе.

Так если глубина страйпа равна 64 КБ, то вычислить ширину страйпа мы можем, умножив это значение на количество стрипов с данными в страйпе.

Коэффициент несбалансированности (imbalance ratio) - метрика, использующаяся для оценки эффективности распределения нагрузки по дискам во время восстановления RAID-массива. Вычисляется для фиксированных номеров отказавших дисков как отношение количества операций ввода вывода для наиболее и наименее загруженных дисков при восстановлении RAID-массива. Для оценки RAID-массива в целом используется минимальное, среднее и максимальное значение коэффициента несбалансированности среди всех возможных отказов дисков.

Карта размещения страйпов (stripe map) - структура, хранящая информацию о том, на каком базовом устройстве (диске RAID-массива) расположен каждый стрип каждого страйпа RAID-массива. Для экономии ресурсов используются карты, длина которых значительно меньше, чем количество стрипов в RAID-массиве, доступе к карте размещения страйпов при этом осуществляется по модулю от деления на длину карты размещения страйпов.

Матрица сочетаний - квадратная таблица чисел с длиной стороны, равной количеству дисков в RAID-массиве. Число в таблице, стоящее в i-м столбце, j-й строке обозначает, сколько раз диск i и диск j встречались в одном страйпе согласно карте размещения страйпов.

Данное техническое решение может быть реализовано на компьютере, в виде автоматизированной информационной системы (АИС), распределенной компьютерной системы, или машиночитаемого носителя, содержащего инструкции для выполнения вышеупомянутого способа размещения данных в RAID-массивах для сбалансированного распределения нагрузки во время восстановления массива с помощью вычислительных средств (например, процессора).

На Фиг. 1 представлена блок-схема выполнения заявленного способа (100) размещения данных в RAID-массивах для сбалансированного распределения нагрузки во время восстановления массива.

На первом этапе (101) создают новый RAID-массив с помощью процедуры генерации карты размещения страйпов (generate stripe map).

На этапе (102) на вход процедуры генерации карты размещения страйпов передают количество свободных дисков N, длину текущего страйпа L и длину карты размещения страйпов R и на основе полученных данных формируют карту размещения страйпов, состоящую из R конкатенированных перестановок множества {1, …, N}.

Карта размещения страйпов - это массив длиной k*N, где N - количество дисков, k - фиксированный коэффициент, в нашем случае равный 1024.

К карте размещения страйпов предъявляются следующие требования:

Карта размещения страйпов может восприниматься как конкатенация k перестановок множества {1,...,N}, что выступает гарантией равномерного использования дисков т.к. каждый диск n при этом встречается ровно k раз.

Карта размещения страйпов может восприниматься как конкатенация (k*N)/M перестановок множества {1,...,M}, где M - длина страйпа, что выступает гарантией правильно сформированного (без повторений) страйпа.

На фигуре 3 приведен пример требования к карте размещения страйпов, где N - количество дисков, M - длина страйпа, d_i - номер диска из множества {1, …, N}, а i - индекс в карте размещения страйпов из множества {1, …, k*N}.

На этапе (103) осуществляют инициализацию матрицы сочетаний M размером NxN, где N - количество дисков, во время которой присваивают всем ее элементам значения 0, текущий страйп инициализируется пустым списком.

Инициализация матрицы сочетаний M - это присваивание всем ее элементам значения 0. Текущий страйп инициализируется пустым списком. Текущий страйп - это название вспомогательной структуры типа список. Элементы списка являются номерами дисков. В разные моменты в списке может находиться от 0 до L (параметр длины страйпа, передаваемый на вход) элементов. Заполнение списка «текущий страйп» происходит в ходе генерации перестановки. Как только его длина достигает L, список сбрасывается. Сам по себе он не сохраняется, а играет роль вспомогательной структуры для генерации перестановки.

Например, элемент матрицы сочетаний M[i][j] отвечает за то, сколько раз диск i и диск j встречались в одном страйпе. Так как отношение нахождения в одном страйпе симметрично, то матрица сочетаний M симметрична относительно своей главной диагонали. В целях оптимизации использования памяти можно хранить только верхнетреугольную часть матрицы сочетаний M. Для простоты изложения в рамках описания алгоритма используется полный вариант матрицы сочетаний M, в котором M[i][j]=M[j][i] для всех i, j из множества {1,...,N}.

Значение коэффициента несбалансированности линейно зависит от отношения минимального и максимального (за исключением элементов на главной диагонали) элементов в матрице сочетаний M.

Представленный выше способ размещения данных составляет карту страйпов пытаясь сбалансировать элементы матрицы сочетаний: напрямую использует текущие данные о сбалансированности сочетаний дисков в страйпах и старается найти лучшую комбинацию на основе этого.

Матрица сочетаний и карта размещения страйпов располагаются в оперативной памяти. При этом матрица сочетаний это временный объект, память под который выделяется только на время заполнения карты размещения страйпов. Карта размещения страйпов же наоборот объект персистентный и располагается в оперативной памяти постоянно. Именно она определяет на каком диске расположен тот или иной блок информации.

На этапе (104) выполняют генерацию перестановки в карте размещения страйпов (фиг.6) путем вызова процедуры генерации перестановки (generate permutation) со следующими входными параметрами: текущее значения матрицы сочетаний M, список занятых дисков в текущем страйпе, длина текущего страйпа L, количество дисков в RAID-массиве N.

Карта размещения страйпов состоит из K конкатенированных перестановок. Перестановка - это произвольный упорядоченный набор всех элементов множества дисков без повторений. Например, перестановками множества {1,2,3} являются

перестановки 1, 2, 3; 3, 1, 2 и др.

Выбор именно такого способа генерации обусловлен тем, что, используя конкатенацию перестановок гарантируется, что все диски встречаются в карте размещения страйпов с одинаковой частотой. В свою очередь, это гарантирует равномерное использование пространства всех дисков.

Передача входных параметров позволяет сгенерировать перестановку, которая с учетом сгенерированных ранее перестановок позволит получить хороший коэффициент несбалансированности для дисков, что также является задачей данного решения. Наиболее важными тут являются матрица сочетаний M и список дисков в текущем страйпе, именно по ним происходит выбор нового диска. Далее этот новый диск добавляется в страйп и в перестановку.

На этапе (105) для генерации одной перестановки на основе входных параметров осуществляют инициализацию вспомогательных структур, а именно: список свободных дисков инициализируется числами от 1 до N, список дисков в текущей перестановке инициализируется пустым списком.

На этапе (106) итеративно осуществляют выбор диска, который содержится в списке свободных дисков, но не содержится в списке занятых дисков в текущем страйпе, используя поиск минимальной суммы элементов матрицы сочетаний, соответствующих диску и занятых дисков в текущем страйпе. Минимальную сумму элементов матрицы сочетаний рассчитывают на этапе генерации перестановки. На каждом шаге в генерации перестановки добавляют один диск. Диск выбирается на основе минимума сумм в матрице сочетаний. Суммируемые элементы при этом выбираются по следующей логике, номером строки всегда является номер диска-кандидата. Номер столбца меняется в цикле, проходящем по всем элементам массива текущего страйпа.

Процедура выбора диска (фиг.7) получает на вход матрицу сочетаний M, список дисков в текущем страйпе S и непустой список доступных дисков A. Первым делом инициализируем вспомогательные переменные, это номер диска с наименьшей локальной суммой и ему присваивается первый элемент списка доступных дисков. Минимум локальных сумм и ему присваивается значение максимального значения типа INT. Далее итеративно проходим по всем элементам списка A. Присваиваем в начале каждой итерации значение локальной суммы, равное нулю. Проходим по всем элементам списка дисков в текущем страйпе S и добавляем к локальной сумме значение матрицы сочетаний M, стоящее в ячейке (значение текущего элемента непустого списка A, значение текущего элемента непустого списка S). после прохода по списку S получаем значение локальной суммы. Если оно меньше значения текущего минимума локальных сумм, то запоминаем номер диска. Процедура возвращает диск из списка A с наименьшим значением локальной суммы.

На этапе (107) добавляют диск в список занятых дисков в текущем страйпе и в конец списка дисков в текущей перестановке.

На этапе (108) как только длина списка занятых дисков в текущем страйпе достигла длины текущего страйпа L, обновляют матрицу сочетаний и присваивают списку занятых дисков в текущем страйпе значение пустого списка и повторяют процедуру генерации перестановки в карте размещения страйпов до тех пор, пока количество перестановок не достигло R.

Процедура обновления матрицы сочетаний (фиг.8) получает на вход матрицу сочетаний и текущий страйп.Итеративно проходим по всем возможным значениям кортежа (d1, d2), где d1 и d2 это номера дисков из списка S и увеличиваем на 1 значение M[d1][d2]. После обработки всех кортежей процедура возвращает обновленную матрицу

сочетаний.

После исполнения всех указанных выше этапов цикла осуществляют возвращение карты размещения страйпов, которая продолжит храниться в оперативной памяти на протяжении всего времени работы RAID-массива.

Как следует из указанного выше, заявленное решение позволяет обеспечить высокий уровень надежности хранения данных и увеличение скорости восстановления RAID-массива за счет схемы расположения данных, обеспечивающей равномерное распределение нагрузки чтения по всем дискам во время восстановления данных.

На Фиг. 9 представлен общий пример вычислительного устройства (900), которое может представлять собой, например, компьютер, сервер, ноутбук, смартфон, SoC (System-on-a-Chip/Система на кристалле) и т.п. Устройство (900) может применяться для полной или частичной реализации заявленного способа (100).

В общем случае устройство (900) содержит такие компоненты, как: один или более процессоров (901), по меньшей мере одну оперативную память (902), средство постоянного хранения данных (903), интерфейсы ввода/вывода (904) включая релейные выходы для соединения с контроллерами управления движения ленточного конвейера, средство В/В (905), средства сетевого взаимодействия (906).

Процессор (901) устройства выполняет основные вычислительные операции, необходимые для функционирования устройства (900) или функционала одного или более его компонентов. Процессор (901) исполняет необходимые машиночитаемые команды, содержащиеся в оперативной памяти (902).

Память (902), как правило, выполнена в виде ОЗУ и содержит необходимую программную логику, обеспечивающую требуемый функционал. Средство хранения данных (903) может выполняться в виде HDD, SSD дисков, рейд массива, сетевого хранилища, флэш-памяти, оптических накопителей информации (CD, DVD, MD, BlueRay дисков) и т.п.Средство (903) позволяет выполнять долгосрочное хранение различного вида информации, например, запись магнитограмм, истории обработки запросов (логов), идентификаторов пользователей, данные камер, изображения и т.п.

Интерфейсы (904) представляют собой стандартные средства для подключения и работы с вычислительными устройствами. Интерфейсы (904) могут представлять, например, релейные соединения, USB, RS232/422/485 или другие, RJ45, LPT, UART, COM, HDMI, PS/2, Lightning, Fire Wire и т.п.для работы, в том числе, по протоколам Modbus и сетям Probfibus, Profinet или сетям иного типа. Выбор интерфейсов (904) зависит от конкретного исполнения устройства (900), которое может представлять собой, вычислительный блок (вычислительный модуль), например на базе ЦПУ (одного или нескольких процессоров), микроконтроллера и т.п., персональный компьютер, мейнфрейм, серверный кластер, тонкий клиент, смартфон, ноутбук и т.п., а также подключаемых сторонних устройств.

В качестве средств В/В данных (905) может использоваться: клавиатура, джойстик, дисплей (сенсорный дисплей), проектор, тачпад, манипулятор мышь, трекбол, световое перо, динамики, микрофон и т.п.

Средства сетевого взаимодействия (906) выбираются из устройства, обеспечивающего сетевой прием и передачу данных, например, Ethernet карту, WLAN/Wi-Fi модуль, Bluetooth модуль, BLE модуль, NFC модуль, IrDa, RFID модуль, GSM модем, и т.п. С помощью средства (906) обеспечивается организация обмена данными по проводному или беспроводному каналу передачи данных, например, WAN, PAN, ЛВС (LAN), Интранет, Интернет, WLAN, WMAN или GSM, квантовый (оптоволоконный) канал передачи данных, спутниковая связь и т.п. Компоненты устройства (900), как правило,

сопряжены посредством общей шины передачи данных.

Программа - последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины или устройством обработки команд.

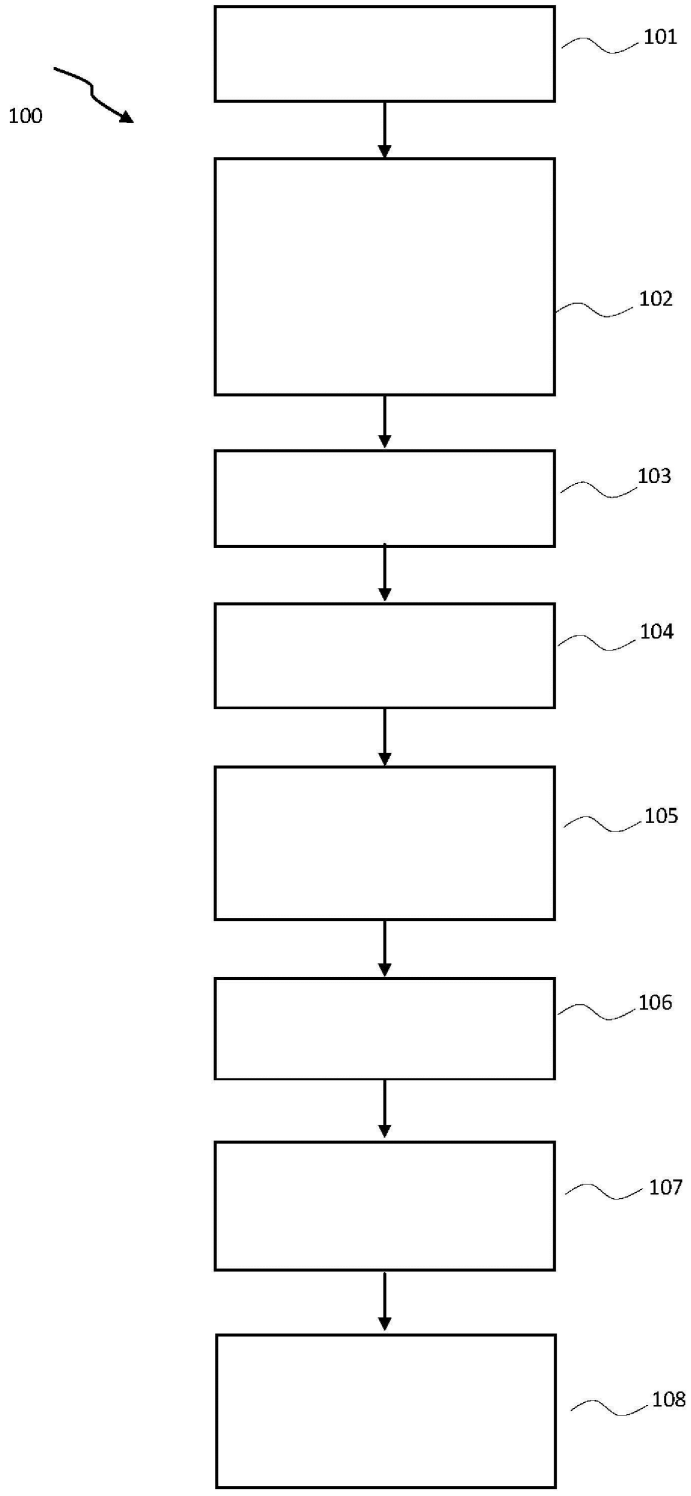
В настоящих материалах заявки было представлено предпочтительное раскрытие осуществления заявленного технического решения, которое не должно использоваться как ограничивающее иные, частные воплощения его реализации, которые не выходят за рамки испрашиваемого объема правовой охраны и являются очевидными для специалистов в соответствующей области техники.

(57) Формула изобретения

Компьютерно-реализуемый способ размещения данных в RAID-массивах для сбалансированного распределения нагрузки во время восстановления массива, содержащий этапы, на которых:

- создают новый RAID-массив с помощью процедуры генерации карты размещения страйпов (generate stripe map);
- на вход процедуры генерации карты размещения страйпов передают количество свободных дисков N, длину текущего страйпа L и длину карты размещения страйпов R и на основе полученных данных формируют карту размещения страйпов, состоящую из R конкатенированных перестановок множества {1, …, N};
- осуществляют инициализацию матрицы сочетаний M размером N×N, во время которой присваивают всем ее элементам значения 0, текущий страйп инициализируется пустым списком;
- выполняют генерацию перестановки в карте размещения страйпов путем вызова процедуры генерации перестановки (generate permutation) со следующими входными параметрами: текущее значение матрицы сочетаний M, список занятых дисков в текущем страйпе, длина текущего страйпа L, количество дисков в RAID-массиве N;
- для генерации одной перестановки на основе входных параметров осуществляют инициализацию вспомогательных структур, а именно: список свободных дисков инициализируется числами от 1 до N, список дисков в текущей перестановке инициализируется пустым списком;
- итеративно осуществляют выбор диска, который содержится в списке свободных дисков, но не содержится в списке занятых дисков в текущем страйпе, используя поиск минимальной суммы элементов матрицы сочетаний, соответствующих диску, и занятых дисков в текущем страйпе;
- добавляют диск в список занятых дисков в текущем страйпе и в конец списка дисков в текущей перестановке;
- как только длина списка занятых дисков в текущем страйпе достигла длины текущего страйпа L, обновляют матрицу сочетаний и присваивают списку занятых дисков в текущем страйпе значение пустого списка;
- повторяют процедуру генерации перестановки в карте размещения страйпов до тех пор, пока количество перестановок не достигло R.

1



Фиг. 1

2

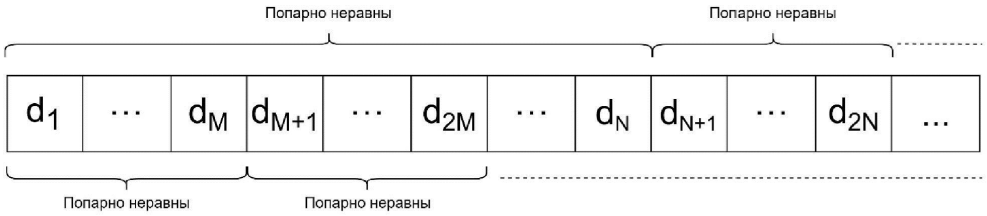
	1	2	3	4	5	6
1	8	3	2	2	1	0
2	3	8	1	1	2	1
3	2	1	8	2	0	2
4	2	1	2	8	1	2
5	1	2	0	1	8	3
6	0	1	2	2	3	8

Матрица сочетаний

Страйп 1			Страйп 2		
1	2	3	4	5	6
Страйп 3			Страйп 4		
2	5	1	3	6	4
Страйп 5			Страйп 6		
3	1	4	6	2	5
Страйп 7			Страйп 8		
4	1	2	3	5	6

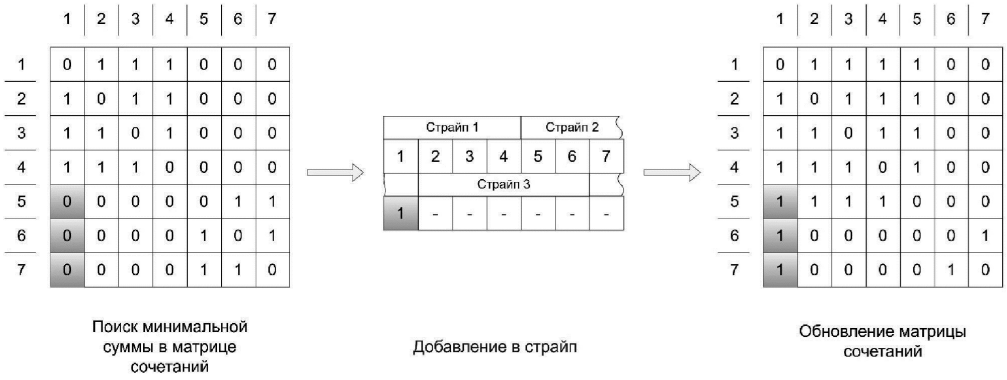
Карта страйпов

Фиг. 2

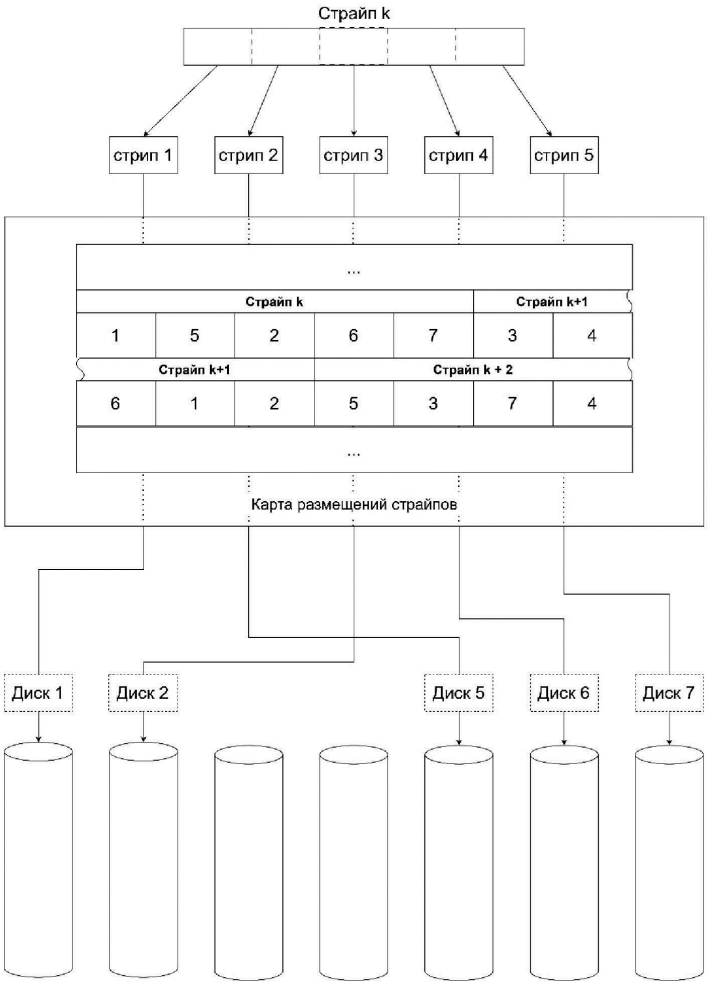


Фиг. 3

3

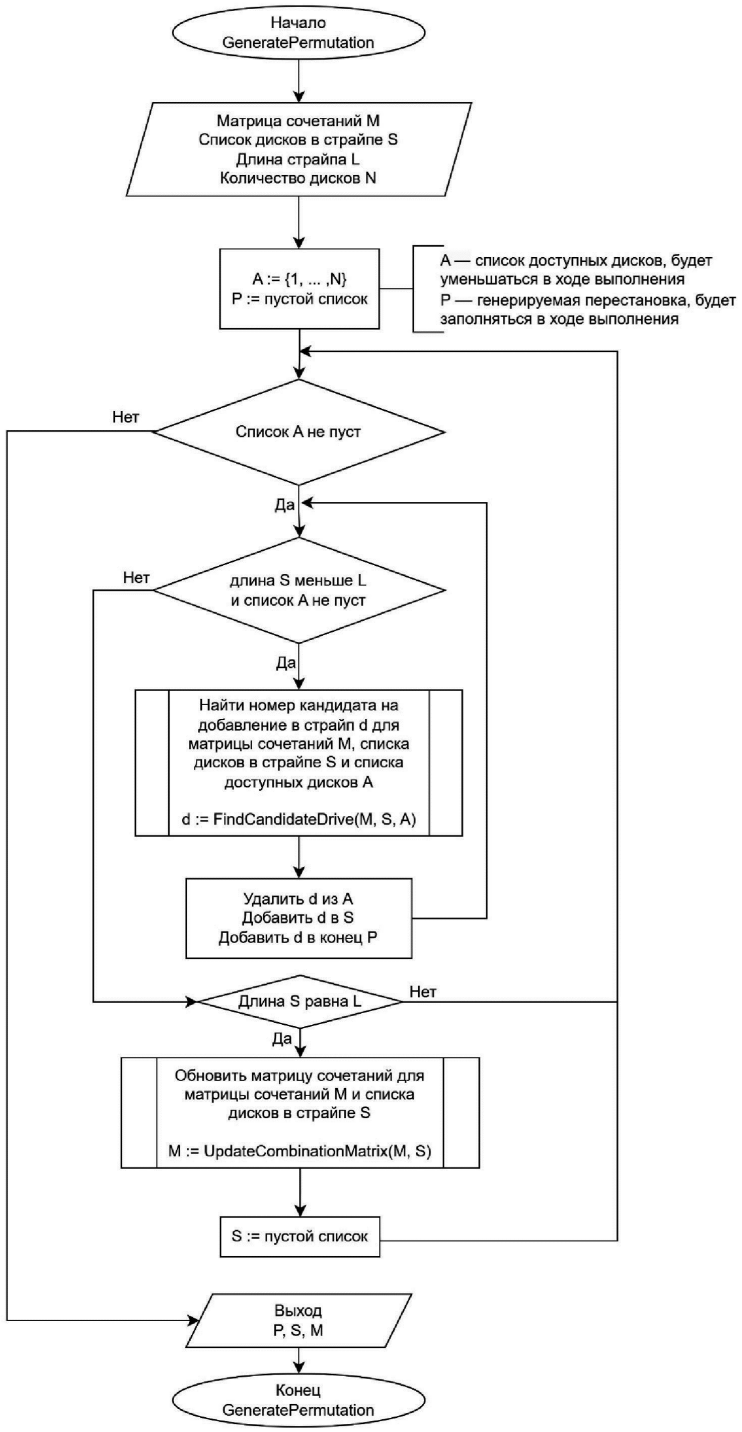

Поиск минимальной суммы в матрице сочетаний
Добавление в страйп
Обновление матрицы сочетаний

Фиг. 4

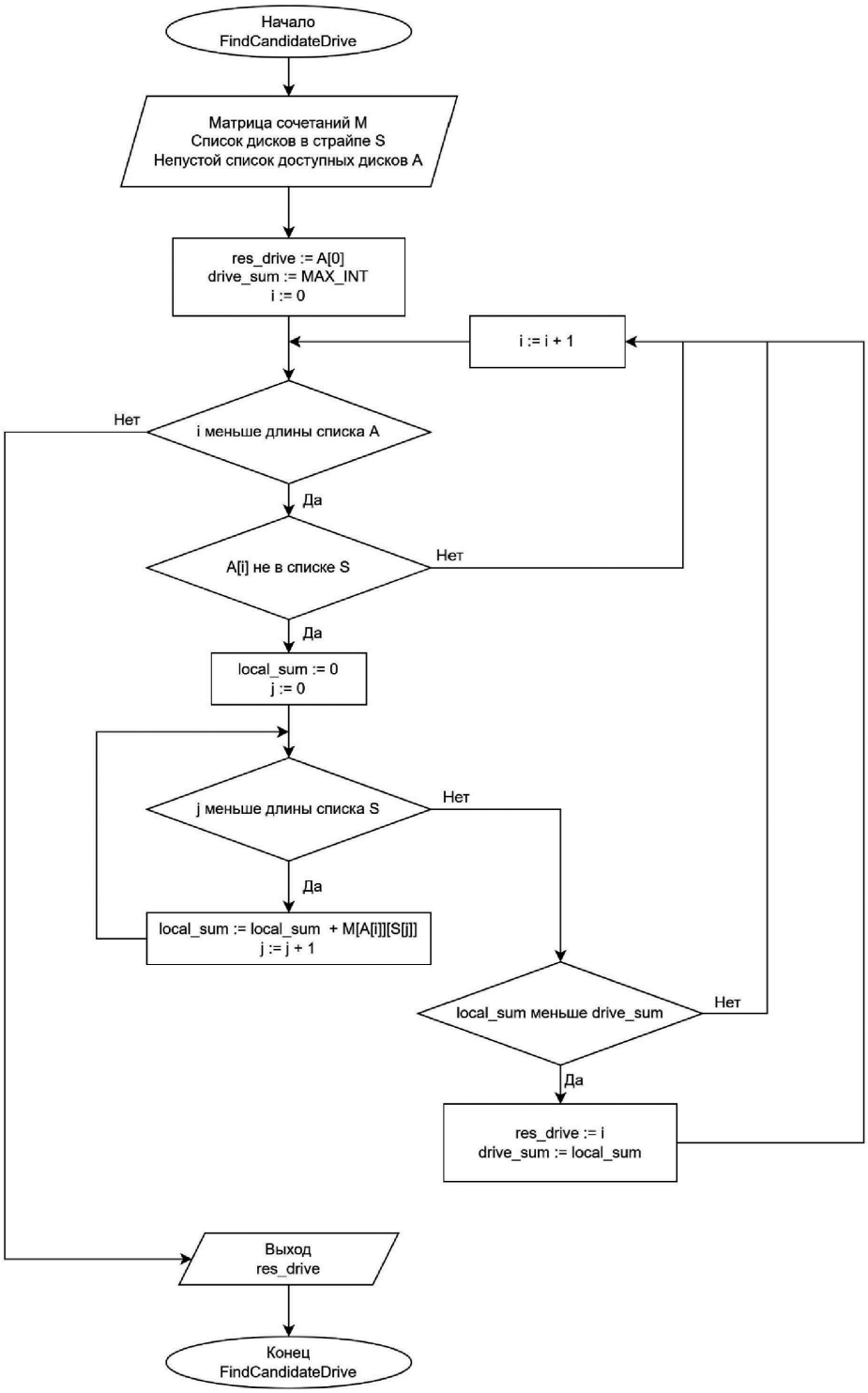

Страйп k
стрип 1
стрип 2
стрип 3
стрип 4
стрип 5
Страйп k
Страйп k+1
Страйп k+1
Страйп k + 2
Карта размещений страйпов
Диск 1
Диск 2
Диск 5
Диск 6
Диск 7

Фиг. 5

4

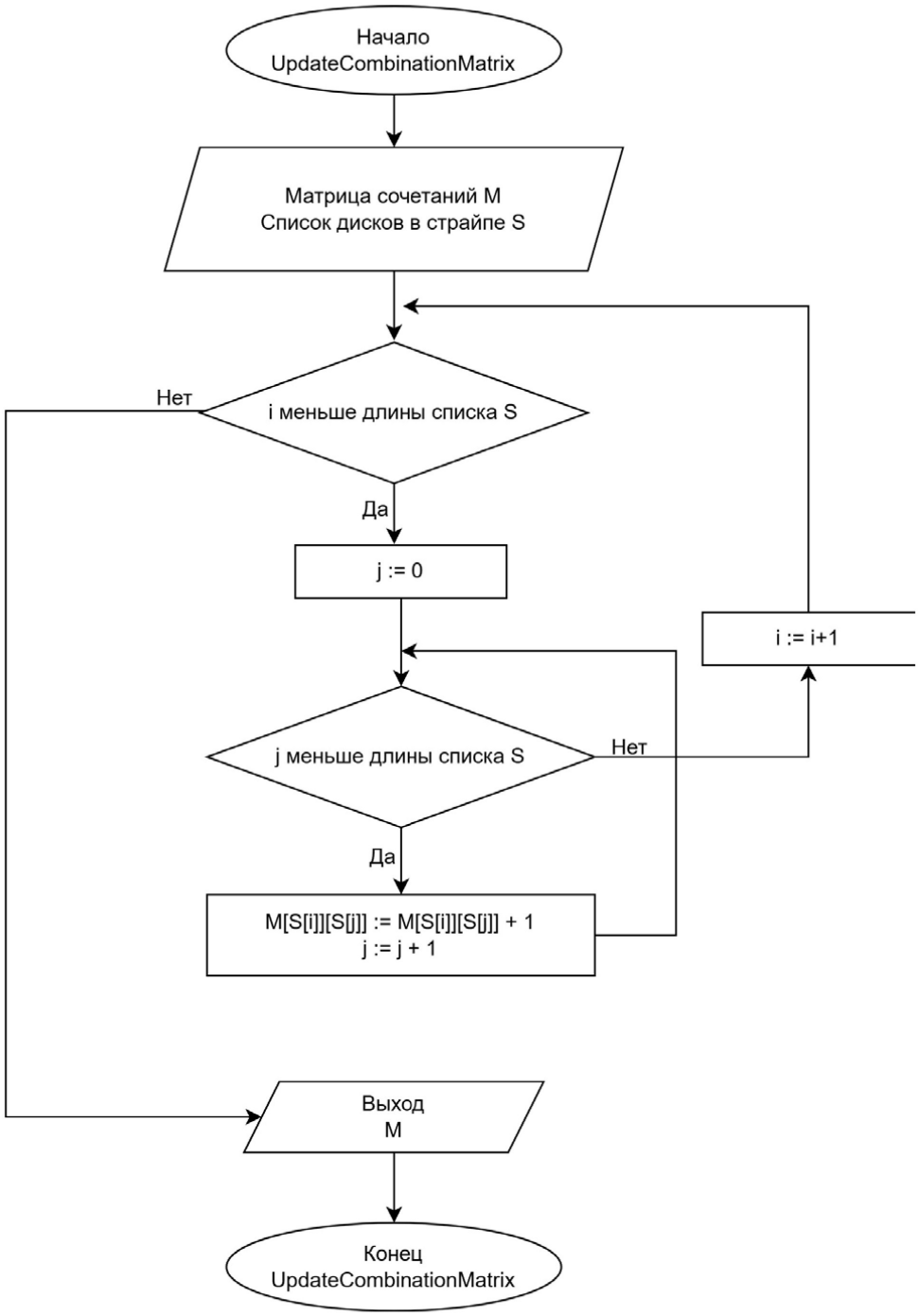

Начало
GeneratePermutation
Матрица сочетаний M
Список дисков в страйпе S
Длина страйпа L
Количество дисков N
A := {1, ... ,N}
P := пустой список
A — список доступных дисков, будет уменьшаться в ходе выполнения
P — генерируемая перестановка, будет заполняться в ходе выполнения
Список A не пуст
Нет
Да
длина S меньше L и список A не пуст
Нет
Да
Найти номер кандидата на добавление в страйп d для матрицы сочетаний M, списка дисков в страйпе S и списка доступных дисков A
d := FindCandidateDrive(M, S, A)
Удалить d из A
Добавить d в S
Добавить d в конец P
Длина S равна L
Нет
Да
Обновить матрицу сочетаний для матрицы сочетаний M и списка дисков в страйпе S
M := UpdateCombinationMatrix(M, S)
S := пустой список
Выход
P, S, M
Конец
GeneratePermutation

Фиг. 6


Начало
FindCandidateDrive
Матрица сочетаний M
Список дисков в страйпе S
Непустой список доступных дисков A
res_drive := A[0]
drive_sum := MAX_INT
i := 0
i := i + 1
i меньше длины списка A
Нет
Да
A[i] не в списке S
Нет
Да
local_sum := 0
j := 0
j меньше длины списка S
Нет
Да
local_sum := local_sum + M[A[i]][S[j]]
j := j + 1
local_sum меньше drive_sum
Нет
Да
res_drive := i
drive_sum := local_sum
Выход
res_drive
Конец
FindCandidateDrive

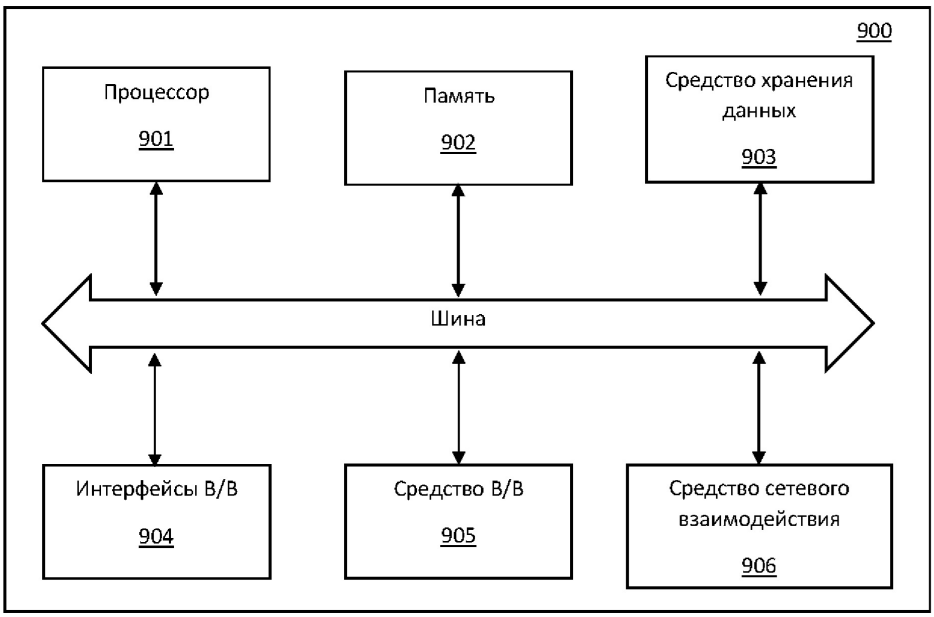
Фиг. 7

6


Начало
UpdateCombinationMatrix
Матрица сочетаний M
Список дисков в страйпе S
i меньше длины списка S
Нет
Да
j := 0
j меньше длины списка S
Нет
Да
M[S[i]][S[j]] := M[S[i]][S[j]] + 1
j := j + 1
i := i+1
Выход
M
Конец
UpdateCombinationMatrix

Фиг. 8

7


900
Процессор
901
Память
902
Средство хранения данных
903
Шина
Интерфейсы В/В
904
Средство В/В
905
Средство сетевого взаимодействия
906

Фиг. 9