



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 12/0246 (2006.01); *G11C 29/808* (2006.01); *G11C 29/82* (2006.01); *G11C 13/004* (2006.01); *G11C 2029/0409* (2006.01); *G11C 13/0004* (2006.01)

(21)(22) Заявка: 2015118415, 15.11.2013

(24) Дата начала отсчета срока действия патента:
15.11.2013Дата регистрации:
06.09.2018

Приоритет(ы):

(30) Конвенционный приоритет:
16.11.2012 US 13/678,941(43) Дата публикации заявки: 10.12.2016 Бюл. №
34

(45) Опубликовано: 06.09.2018 Бюл. № 25

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 15.05.2015(86) Заявка РСТ:
US 2013/070365 (15.11.2013)(87) Публикация заявки РСТ:
WO 2014/078695 (22.05.2014)Адрес для переписки:
129090, Москва, ул. Б. Спасская, 25, стр. 3, ООО
"Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

ШТРАУСС Карин (US),
СМИТ Бертон Дж. (US),
МАККИНЛИ Кэтрин С. (US)

(73) Патентообладатель(и):

МАЙКРОСОФТ ТЕКНОЛОДЖИ
ЛАЙСЕНСИНГ, ЭлЭлСи (US)(56) Список документов, цитированных в отчете
о поиске: US 2007/0103977 A1, 10.05. 2007.
WO 2011/019602 A2, 17.02.2011. WO 2011/
005663 A1, 13.01.2011. US 5535328 A,
09.07.1996. RU 2115160 C1, 10.07.1998.(54) ПЕРЕРАСПРЕДЕЛЕНИЕ СЕГМЕНТОВ ПАМЯТИ ДЛЯ РЕШЕНИЯ ПРОБЛЕМЫ
ФРАГМЕНТАЦИИ

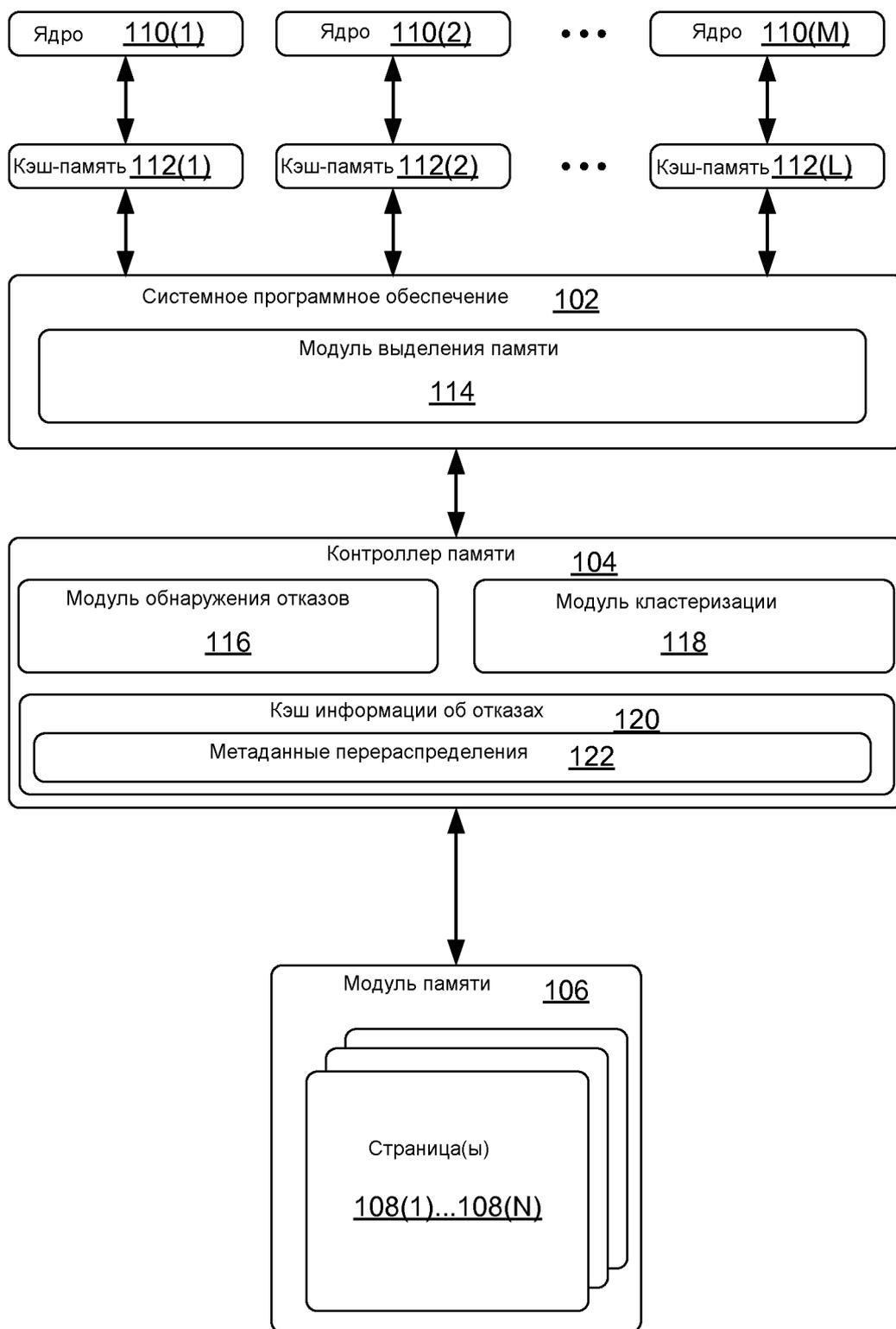
(57) Реферат:

Группа изобретений относится к области запоминающих устройств. Техническим результатом является увеличение надежности хранения данных в устройствах памяти. Система содержит аппаратный модуль памяти, включающий в себя множество областей памяти, причем отдельные области памяти включают в себя множество сегментов; модуль обнаружения отказов, который обнаруживает вышедший из

строю сегмент в области памяти из данного множества областей памяти; модуль кластеризации, который логически кластеризует вышедший из строя сегмент посредством перераспределения вышедшего из строя сегмента в рабочий сегмент, расположенный на краю упомянутой области памяти, причем рабочий сегмент выбирается для максимизации непрерывного пространства памяти в пределах

данной области памяти. 3 н. и 17 з.п. ф-лы, 5 ил.

100 ↘



ФИГ.1

RU 2666241 C2

RU 2666241 C2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.
G06F 11/20 (2006.01)
G11C 29/04 (2006.01)

(12) **ABSTRACT OF INVENTION**

(52) CPC

G06F 12/0246 (2006.01); *G11C 29/808* (2006.01); *G11C 29/82* (2006.01); *G11C 13/004* (2006.01); *G11C 2029/0409* (2006.01); *G11C 13/0004* (2006.01)

(21)(22) Application: **2015118415**, 15.11.2013

(24) Effective date for property rights:
15.11.2013

Registration date:
06.09.2018

Priority:

(30) Convention priority:
16.11.2012 US 13/678,941

(43) Application published: 10.12.2016 Bull. № 34

(45) Date of publication: 06.09.2018 Bull. № 25

(85) Commencement of national phase: 15.05.2015

(86) PCT application:
US 2013/070365 (15.11.2013)

(87) PCT publication:
WO 2014/078695 (22.05.2014)

Mail address:
129090, Moskva, ul. B. Spasskaya, 25, str. 3, OOO
"Yuridicheskaya firma Gorodisskij i Partnery"

(72) Inventor(s):

**SHTRAUSS Karin (US),
SMIT Berton Dzh. (US),
MAKKINLI Ketrin S. (US)**

(73) Proprietor(s):

**MAJKROSOFT TEKNOLODZHI
LAJSENSING, EIEISi (US)**

(54) **MEMORY SEGMENT REMAPPING TO ADDRESS FRAGMENTATION**

(57) Abstract:

FIELD: computing; counting.

SUBSTANCE: group of inventions relates to memory devices. System comprises a hardware memory module including a plurality of memory regions, wherein separate memory regions include a plurality of segments; a failure detection module that detects a failed segment in one of the plurality of memory regions; a clustering module that logically clusters the

failed segment by remapping the failed segment to a working segment located on the edge of said memory region, wherein the working segment is selected to maximize a contiguous memory space within a given memory region.

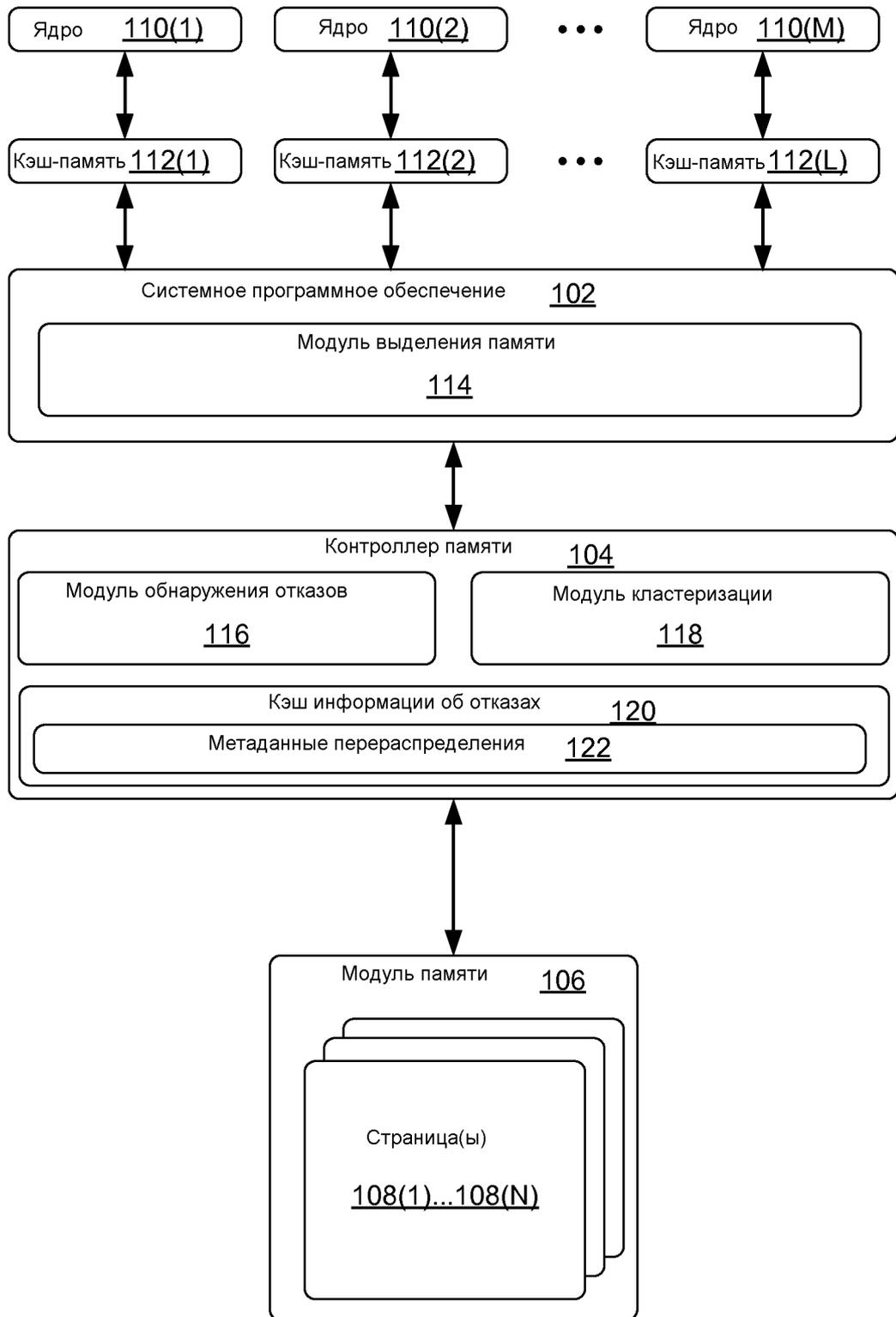
EFFECT: technical result is high reliability of data storage in memory devices.

20 cl, 5 dwg

RU 2 666 241 1 C2

RU 2 666 241 C2

100 ↘



ФИГ.1

RU 2 6 6 6 2 4 1 C 2

RU 2 6 6 6 2 4 1 C 2

УРОВЕНЬ ТЕХНИКИ

В связи с проблемами масштабирования современные технологии могут перейти от использования устройств памяти на основе электрического заряда, таких как динамическое оперативное запоминающее устройство (DRAM), к резистивным и магниторезистивным устройствам памяти, таким как память на основе фазового перехода (PCM). Однако многие такие устройства памяти имеют тенденцию к изнашиванию.

Память может изнашиваться тогда, когда система записывает в память, так как запись может изменять физическую конфигурацию материала в памяти или ухудшать его некоторым другим образом. Устройства памяти, восприимчивые к этой проблеме, упоминаются как изнашиваемые устройства памяти. Из-за постепенного ухудшения параметров изнашиваемые устройства памяти подвергаются отказам памяти, которые могут воспрепятствовать надежному хранению в памяти данных для системы после большого количества операций записи, производимых системой. Например, ячейка PCM может, как правило, выдерживать порядка 10^8 операций записи перед изнашиванием (то есть перед отказом), тогда как ячейка DRAM может, как правило, выдерживать порядка 10^{15} операций записи перед отказом.

Память для системы можно разбить на множество областей, причем каждая область имеет многочисленные сегменты. Система может использовать области памяти в качестве блока для выделения памяти (например, во время исполнения) и/или переноса данных между основной памятью и вспомогательной памятью, такой как жесткий диск. Так как отдельные сегменты областей памяти начинают выходить из строя из-за большого количества записей, происходит фрагментация памяти. Фрагментация памяти возникает в тех случаях, когда область памяти разделена на меньшие участки используемой памяти (например, один или более соседних "рабочих" сегментов), которые разделятся вышедшим из строя или изношенным сегментом, который нельзя больше использовать. Хотя малые участки используемой памяти между вышедшими из строя сегментами являются по-прежнему надежными для выделения памяти, они не используются эффективным образом в системе, так как малые участки по отдельности являются слишком малыми для хранения более крупных объектов, например, данных, чье хранение требует непрерывной памяти (например, многочисленные смежные сегменты). Другими словами, система не может поместить более крупные объекты между двумя вышедшими из строя сегментами памяти из-за фрагментации.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Технологии, которые обсуждаются в данном документе, позволяют увеличить срок службы изнашиваемых устройств памяти за счет решения проблемы фрагментации. Технологии реализованы в объединенной аппаратной и программной системе, которая идентифицирует вышедшие из строя сегменты памяти в области памяти. Технология позволяет в дальнейшем управлять вышедшими из строя сегментами памяти путем логической кластеризации вышедших из строя сегментов памяти на участке, находящемся за пределами области памяти, с использованием процесса перераспределения. Процесс перераспределения может включать в себя создание и сохранение метаданных перераспределения, определяющих информационные записи перераспределения сегментов в области памяти. Соответственно, кластеризация отказов логически исключает или уменьшает фрагментацию памяти таким образом, чтобы система могла выделять более крупные участки непрерывной памяти для хранения объектов.

По меньшей мере в одном варианте осуществления программное обеспечение системы

(например, операционная система) и/или контроллер памяти могут в дальнейшем осуществлять доступ к, использовать и обновлять метаданные перераспределения во время выделения памяти, считываний из памяти, записи в память или их комбинации. Например, программное обеспечение системы или контроллер памяти может выделять память во время исполнения путем сообщения об отказах памяти и указания величины пространства непрерывной памяти в пределах области памяти после перераспределения и кластеризации отказов памяти.

Это краткое изложение сущности изобретения предоставлено, чтобы ввести подборку концепций в упрощенной форме, которые дополнительно описаны ниже в подробном описании. Данное краткое изложение сущности изобретения не предназначено ни для идентификации ключевых или существенных признаков заявляемого изобретения, ни для использования в качестве помощи при определении объема заявляемого изобретения. Например, термин "технологии" может относиться к системе(ам), аппаратным средствам, способу(ам), машиночитаемым инструкциям, модулю(ям) и/или алгоритмам, как это разрешено вышеупомянутым контекстом и на всем протяжении документа.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Подробное описание представлено со ссылкой на сопроводительные фигуры. На фигурах левая крайняя цифра(ы) ссылочной позиции идентифицирует фигуру, на которой ссылочная позиция появляется первой. Одинаковые ссылочные позиции на различных фигурах используются для обозначения аналогичных или идентичных элементов.

Фиг. 1 иллюстрирует примерную архитектуру системы для реализации технологии кластеризации и перераспределения в соответствии с различными вариантами осуществления, раскрытыми в данном документе.

Фиг. 2 иллюстрирует примерную область памяти на различных этапах в случае, когда примерная архитектура системы реализует технологию кластеризации и перераспределения в соответствии с различными вариантами осуществления, раскрытыми в данном документе.

Фиг. 3 иллюстрирует две последовательных примерных области памяти в соответствии с различными вариантами осуществления, описанными в данном документе.

Фиг. 4 иллюстрирует примерный процесс выделения памяти, который создает и/или обновляет метаданные перераспределения в соответствии с различными вариантами осуществления, описанными в данном документе.

Фиг. 5 иллюстрирует примерный процесс выделения памяти, который использует метаданные перераспределения, чтобы считывать данные из области памяти в соответствии с различными вариантами осуществления, описанными в данном документе.

ПОДРОБНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

Обзор

В приведенном ниже описании изложены технологии, которые позволяют логически уменьшить и/или устранить фрагментацию в пределах области памяти, которая получилась в результате вышедших из строя сегментов. Когда система обнаруживает выход из строя сегмента в пределах области памяти, технологии логически перераспределяют отказ на участок, расположенный за пределами области памяти (например, верхняя или нижняя часть области памяти). Технологии позволяют также создавать метаданные перераспределения, показывающие переключения местоположений сегментов и смещения сегментов, с тем чтобы система могла выделить более крупные участки непрерывной памяти, которые получаются в результате

перераспределения вышедших из строя сегментов, участку, расположенному за пределами области памяти. По меньшей мере в одном варианте осуществления метаданные перераспределения являются уникальными для области памяти, и технологии хранят метаданные перераспределения в пределах области памяти. Система
5 может в дальнейшем использовать метаданные перераспределения во время процесса выделения памяти.

Как описано выше, изнашиваемые устройства памяти подвержены отказам памяти, которые могут воспрепятствовать надежному хранению в памяти данных после
большого количества записей. Отказы памяти могут представлять собой изначально
10 присутствующие отказы или статические отказы, либо отказы памяти могут представлять собой динамические отказы. В различных вариантах осуществления память представляет собой основную память, которая может включать в себя большой участок изнашиваемой памяти (которая, как правило, предполагается более плотной и более эффективной в плане затрат памятью) и меньший участок DRAM, который
15 хранит данные и операции, которые не могут допускать отказа (например, операции, которые могут быть существенными для операционной системы). Изнашиваемые устройства памяти могут обеспечивать стабильные сегменты хранения, которые выходят из строя с меньшей степенью детализации по сравнению с более крупными областями основной памяти. В контексте данного документа более крупные области памяти
20 описаны в качестве страниц памяти, и сегменты страниц памяти упоминаются как строки. Однако, в контексте данного документа понятно, что области памяти могут представлять собой любой модуль или блок памяти, используемый для выделения памяти в изнашиваемой памяти или устройствах памяти, выполненных по другой технологии. Более того, области памяти и строки могут изменяться по размеру или
25 объему для хранения объектов (например, области и/или строки могут быть более детализированными или менее детализированными).

Когда в системе, которая использует DRAM в качестве основной памяти, происходит отказ памяти, отключается вся область памяти (например, страница). Однако
продолжительный отказ памяти, где используется DRAM, является очень редким.
30 Напротив, в изнашиваемых устройствах памяти продолжительные отказы памяти происходят чаще. Если бы одинаковое или аналогичное выведение из строя всей области памяти проявлялось бы для изнашиваемых устройств памяти, то объем памяти быстро бы израсходовался даже в том случае, если бы каждая область памяти по-прежнему имела бы все, а не один функциональный сегмент. По этой причине, вместо отключения
35 памяти с детализацией на уровне области, системы, использующие изнашиваемые устройства памяти, могут отключать память с меньшей степенью детализации на уровне сегмента (например, с детализацией на уровне строки). Однако, так как трансляцию памяти из виртуальных адресов в физические адреса можно все еще выполнить с детализацией на уровне области в системах с изнашиваемой памятью (и это обычно
40 выполняется при детализациях на уровне областей по причинам непроизводительных затрат), это может привести к виртуальному адресному пространству, которое нельзя сделать полностью непрерывным, и, следовательно, к фрагментации.

Таким образом, технологии, обсуждаемые в данном документе, обращают внимание на вышеупомянутые проблемы фрагментации памяти путем кластеризации вышедших
45 из строя строк памяти на участке, находящемся за пределами отдельных страниц памяти (например, верхняя или нижняя часть страниц памяти). Для того чтобы выполнить кластеризацию вышедших из строя строк памяти, способы позволяют реализовать процесс перераспределения в пределах каждой страницы памяти. В результате,

предложенной технологией логически создаются более крупные участки непрерывной памяти, поэтому объекты, более крупные, чем рабочая память между любыми двумя вышедшими из строя сегментами, можно по-прежнему хранить на странице, когда начинают происходить отказы сегментов. Это увеличивает срок службы области.

5 В некоторых вариантах осуществления способы позволяют выполнить кластеризацию вышедших из строя строк памяти на участках, находящихся за пределами большей области перераспределения памяти. Например, способы можно выполнить с
возможностью размещения вышедших из строя строк памяти на верхней или нижней
10 части каждой второй, четвертой, восьмой и т.д. страниц памяти вместо каждой отдельной
страницы памяти. Поэтому эти технологии можно выполнить с возможностью
реализации процессов перераспределения и кластеризации в отношении
"перераспределяемой" области памяти, которая кратна размеру области модуля памяти
аппаратных средств (например, двум страницам памяти, четырем страницам памяти,
15 восьми страницам памяти и т.д.). Это позволяет дополнительно исключить
фрагментацию и создать более крупные участки непрерывных блоков хранения.

В различных вариантах осуществления процесс кластеризации и перераспределения отказов, описанный в данном документе, можно реализовать после того, как система
использует и израсходует другие ресурсы исправления ошибок памяти. Например,
20 другие ресурсы исправления ошибок памяти могут включать в себя стандартные
технологии исправления ошибок, такие как код с исправлением ошибок (ECC) (которые,
как правило, используются в настоящее время для DRAM) и информационные записи
указателей исправления ошибок.

Иллюстративная архитектура

На Фиг. 1 изображена примерная системная архитектура 100 для реализации
25 технологий, которые обсуждаются в данном документе. Системная архитектура 100
может быть сконфигурирована в виде любого подходящего вычислительного
устройства, которая использует технологии памяти, которые допускают фрагментацию.
В качестве примера, а не ограничения, системная архитектура 100 может включать в
себя персональные вычислительные устройства, портативные вычислительные
30 устройства, смартфоны, серверные вычислительные устройства, группы серверов,
центры обработки данных или любые другие устройства, которые используют и/или
выделяют память. Соответственно, системная архитектура 100 может включать в себя
системное программное обеспечение 102, контроллер 104 памяти и модуль 106 памяти.
Системное программное обеспечение 102, контроллер 104 памяти и/или модуль 106
35 памяти могут быть соединены и могут осуществлять связь в виде множества различных
конфигураций (например, системная шина, сеть и т.д.).

В различных вариантах осуществления системное программное обеспечение 102
может включать в себя операционную систему, такую как операционная система
Microsoft Windows®, операционная система смартфона, операционная система,
40 работающая в режиме реального времени, или любой другой тип операционной системы,
соединенный с модулем 106 памяти. Более того, системное программное обеспечение
102 может включать в себя любой тип динамических исполняемых сред (например,
виртуальные машины, управляемый код и т.д.). По меньшей мере в одном варианте
осуществления операционная система может сообщать об отказах памяти в
45 динамические исполняемые среды, с тем чтобы динамически исполняемые среды могли
выделять память в окрестности вышедших из строя частей памяти.

В различных вариантах осуществления системное программное обеспечение 102 и/
или контроллер 104 памяти выполнены с возможностью управления потоком данных

в и из модуля 106 памяти (например, физической памяти). Модуль 106 памяти может представлять собой аппаратный модуль памяти, который включает в себя множество страниц 108(1)...108(N), где каждая страница 108(1)...108(N) может представлять собой более крупную область памяти, обсужденную выше. В связи с этим системное программное обеспечение 102 и/или контроллер 104 памяти можно выполнить с возможностью выделения пространства памяти на страницах 108(1)...108(N) для объектов, ассоциированных с программами или приложениями, исполняемыми на одном или более процессорных ядрах 110(1)...110(M), управление которыми осуществляется с помощью системного программного обеспечения 102. Следовательно, системное программное обеспечение 102 можно выполнить с возможностью предоставления сервисов одному или более процессорным ядрам 110(1)...110(M), и каждое процессорное ядро 110(1)...110(M) может иметь кэш 112(1)...112(L), в который кэшируются одна или более страниц 108(1)...108(N), которые в текущий момент времени доступны системному программному обеспечению 102 и/или контроллеру 104 памяти. Более того, в кэшах 112(1)...112(L) может также кэшироваться информация об отказах памяти или ошибках, которая сообщается с помощью системного программного обеспечения 102. По меньшей мере в одном варианте осуществления кэши 112(1)...112(L) могут представлять собой статическое оперативное запоминающее устройство (SRAM).

В различных вариантах осуществления модуль 106 памяти представляет собой основную память для системной архитектуры 100 системного программного обеспечения 102. Системное программное обеспечение 102 можно хранить в модуле 106 памяти или можно хранить в отдельной памяти в пределах системной архитектуры 100, которая отличается от модуля 106 памяти. Контроллер 104 памяти может представлять собой часть аппаратных средств компьютера, которые расположены на печатной плате с модулем 106 памяти, контроллер 104 памяти может быть расположен на той же самой подложке, что и модуль 106 памяти, или контроллер 104 памяти может быть расположен на подложке микросхемы с процессорными ядрами 110(1)...110(M). По меньшей мере в одном варианте осуществления контроллер 104 памяти встроен в программно-аппаратные средства (firmware) или в программируемую пользователем вентильную матрицу (FPGA). В другом варианте осуществления контроллер может представлять собой специализированную интегральную схему (ASIC), ассоциированную с модулем 106 памяти, или драйвер устройства, ассоциированный с модулем 106 памяти. Таким образом, контроллер 104 памяти может располагаться на стороне обработки системной архитектуры 100, или контроллер 104 памяти может локально располагаться на модуле 106 памяти.

Как описано выше, каждая из страниц 108(1)...108(N) включает в себя многочисленные строки, которые системное программное обеспечение 102 и/или контроллер 104 памяти могут использовать для выделения памяти. Размер страницы памяти и число страниц памяти в модуле памяти может изменяться в зависимости от системной архитектуры. Например, размер страницы памяти может находиться в диапазоне от приблизительно одного килобайта до двух мегабайтов. Более того, по меньшей мере в одном примере, каждая строка на странице памяти может иметь размер 512 битов, однако в других примерах страницы и строки могут быть меньше или больше.

В различных вариантах осуществления каждое из процессорных ядер 110(1)...110(M) может включать в себя один или более процессоров. Процессоры могут представлять собой одиночное процессорное устройство или ряд из двух или более устройств, любые или все из которых могут включать в себя многочисленные вычислительные устройства.

Процессоры можно реализовать в виде одного или более микропроцессоров, микрокомпьютеров, микроконтроллеров, процессоров для цифровой обработки сигналов, центральных процессорных устройств, конечных автоматов, логических схем, процессоров с общей памятью и/или любых устройств, которые манипулируют сигналами на основании рабочих инструкций. Среди других возможностей, процессоры можно выполнить с возможностью извлечения и исполнения машиночитаемых инструкций, которые хранятся в системном программном обеспечении 102, контроллере 104 памяти или любом другом приложении или программе, доступной системной архитектуре 100.

Как дополнительно обсуждено здесь, страницы памяти 108(1)...108(N) в модуле 106 памяти могут представлять собой страницы изнашиваемой памяти, такой как РСМ. Однако технология, описанная в данном документе, широко применима к системам, использующим другие типы технологий памяти аппаратных средств, включая, но не ограничиваясь, флэш-память, подсистемы RAM, магниторезистивное оперативное запоминающее устройство с переносом спинового момента (STT-MRAM), ферроэлектрическое оперативное запоминающее устройство (FRAM), резистивное оперативное запоминающее устройство (RRAM) или любые другие магниторезистивные устройства памяти. Более того, технологии, обсужденные в данном документе, могут быть также применимы к устройству памяти на основе заряда, таким как DRAM.

В различных вариантах осуществления системное программное обеспечение 102 может включать в себя модуль 114 выделения памяти, и контроллер 104 памяти может включать в себя модуль 116 обнаружения отказов, модуль 118 кластеризации и кэш 120 информации об отказах. Модуль 114 выделения памяти можно выполнить с возможностью управления потоком данных (например, объектов), поступающих на и из страниц 108(1)...108(N) в модуле 106 памяти. Модуль 116 обнаружения отказов можно выполнить с возможностью обнаружения вышедших из строя строк в пределах каждой из страниц 108(1)...108(N). Например, модуль 116 обнаружения отказов обнаруживает и подтверждает, что строка вышла из строя и не является больше надежной в случае, когда контроллер 104 памяти записывает данные в строку в странице 108(N), к которой в текущий момент времени осуществляется доступ, при одновременном временном сохранении другой версии этих данных в буфере. Затем контроллер 104 памяти считывает данные обратно (например, непосредственно считывает данные обратно) со страницы 108(N), к которой в текущий момент времени осуществляется доступ, и модуль 116 обнаружения отказов сравнивает считанные данные с другой версией данных, которые временно хранятся в буфере. Если данные, считанные обратно из страницы 108(N), к которой в текущий момент времени осуществляется доступ, не являются такими же, как данные, временно хранящиеся в буфере, то модуль 116 обнаружения отказов идентифицирует строку в пределах страницы 108(N), к которой в текущий момент времени осуществляется доступ, в качестве вышедшей из строя строки.

В различных вариантах осуществления модуль 116 обнаружения отказов можно выполнить с возможностью сохранения информации об отказах области в кэше 120 информации об отказах. Информация об отказах областей включает в себя то, имеет или не имеет каждая из отдельных страниц 108(1)...108(N) постоянный отказ строки. Если страница 108(N) имеет постоянный отказ строки, контроллер 104 памяти может кэшировать метаданные 122 перераспределения для страницы 108(N) в кэше 120 информации об отказах, а также адрес для страницы 108(N). Соответственно, модуль 114 выделения памяти может использовать метаданные 122 перераспределения, которые

хранятся в кэше 120 информации об отказах, для управления запросами на считывание и/или запись в отношении страниц 108(1)...108(N), которые имеют по меньшей мере один постоянных отказ строки.

5 Как дополнительно обсуждено ниже, в различных вариантах осуществления модуль 118 кластеризации выполнен с возможностью создания метаданных 122 перераспределения, которые переназначают вышедшие из строя строки краям каждой из отдельных страниц 108(1)...108(N) (например, верхней части страницы или нижней части страницы) или краям страницы 108(1)...108(N), которая находится в конце более крупной области перераспределения памяти (например, четыре страницы или восемь страниц). В некоторых вариантах осуществления модуль 118 кластеризации может 10 создавать метаданные 122 перераспределения для отдельной страницы 108(N) или более крупную область перераспределения памяти, когда модуль 116 обнаружения отказов первым идентифицирует постоянный отказ строки (то есть вышедшую из строя строку) в пределах отдельной страницы 108(N).

15 В различных вариантах осуществления, когда страница 108(N) используется в текущий момент времени, предполагается к использованию или недавно использовалась системным программным обеспечением 102 совместно с одним или более процессорными ядрами 110(1)...110(M), модуль 114 выделения памяти может временно хранить метаданные 122 перераспределения для текущей страницы 108(N) в кэш 120 информации об отказах, как будет дополнительно обсуждено здесь. В альтернативных вариантах осуществления, метаданные 122 перераспределения будут также храниться в одном или более кэшах 112(1)...112(L).

Хотя модуль 116 обнаружения отказов и модуль 118 кластеризации изображены в контроллере 104 памяти, в контексте данного документа понятно, что некоторые или 25 все функциональные возможности, выполняемые этими отдельными модулями, можно дополнительно или альтернативно сохранить где-нибудь еще или расположить на или распространить через системное программное обеспечение 112 (например, некоторые функциональные возможности можно выполнить с помощью операционной системы). Более того, в различных вариантах осуществления системное программное обеспечение 102 может также поддерживать отдельный кэш соотнесения отказов, в котором хранится информация об отказах памяти и который передает эту информацию в динамические исполняемые среды в точках выделения памяти, с тем чтобы можно было избежать отказа памяти. В результате, по меньшей мере в некоторых вариантах осуществления системное программное обеспечение 102 и контроллер 104 памяти могут осуществлять 35 связь друг с другом и обмениваться информацией для выполнения технологий, обсуждаемых в данном документе.

Фиг. 2 иллюстрирует различные стадии для отдельной страницы 108(N) памяти, когда страница памяти изнашивается, и операционная система 102 и контроллер 104 памяти реализуют примерный процесс перераспределения во времени. Как обсуждено выше, 40 отдельная страница 108(N) памяти может включать в себя многочисленные различные строки. Объем памяти каждой строки и число строк на каждой странице может зависеть от типа используемой памяти. По меньшей мере в одном варианте осуществления размер строки может составлять 512 битов, и страница 108(N) может представлять собой страницу размером четыре килобайта.

45 Однако, как показано на Фиг. 2, страница 108(N) включает в себя восемь строк, индексированных 0, 1, 2, 3, 4, 5, 6 и 7 в иллюстративных целях и для облегчения понимания. Хотя строки на Фиг. 2 проиндексированы последовательно с использованием числовых значений от 0 до 7, в контексте документа понятно, что

можно использовать любую схему адресации сегментов или строк для более крупных областей памяти. Более того, как показано на Фиг. 2, каждую строку можно разбить на участок хранения данных и участок исправления ошибок. Участок хранения данных хранит объекты или части объектов, выделенных модулю 106 памяти, и участок исправления ошибок зарезервирован для возможного исправления ошибок для каждой отдельной строки. Когда исправление ошибок полностью исчерпано (например, участок исправления ошибок полностью заполнен на Фиг. 2), модуль 116 обнаружения идентифицирует строку как вышедшую из строя строку, как обсуждено выше. В альтернативных вариантах осуществления отдельные строки могут не включать в себя участок исправления ошибок.

Как показано на этапе 200(A), страница 108(N) является идеальной страницей, так как она еще не включает в себя каких-либо вышедших из строя строк, и каждая строка имеет полный набор доступных ресурсов исправления ошибок. Однако, так как системное программное обеспечение 102 и/или контроллер 104 памяти начинают выполнять запись 108(N) во время процессов выделения памяти, проводимых с течением времени, и из-за изменения в пределах отдельных ячеек, которые образует изнашиваемая память, отдельные строки в пределах страницы 108(N) могут иметь различный уровень износа.

Как показано на этапе 200(B), строка 1 и строка 3, помеченные "отказ 1" и "отказ 2", соответственно, исчерпали все доступные ресурсы исправления ошибок. Таким образом, модуль 116 обнаружения отказов может обнаружить, что эти строки вышли из строя, и в целях этого обсуждения предполагается, что строка 1 вышла из строя первой, а строка 3 вышла из строя второй после выхода из строя строки 1.

В различных вариантах осуществления, когда модуль 116 обнаружения отказов обнаруживает, что первая строка (например, строка 1) в пределах отдельной страницы 108(N) вышла из строя, модуль 118 кластеризации на этапе 200(C) создает метаданные 122 перераспределения и выделяет строку памяти (например, строку 7) в пределах страницы 108(N) для метаданных 122 перераспределения. Таким образом, когда первая строка 108(N) выходит из строя в результате износа (например, строка 1) модуль 118 кластеризации может также создать в то же самое время второй "ложный" отказ. Вторая строка с ложным отказом, которая хранит метаданные 122 перераспределения, может быть расположена на участке за пределами страницы 108(N) (например, строка 7). Модуль 118 кластеризации устанавливает метаданные 122 перераспределения на верхней части или на нижней части страницы 108(N), с тем чтобы не прерывалось непрерывное пространство для хранения данных.

В различных вариантах осуществления метаданные 122 перераспределения включают в себя ряд информационных записей, которые отражают число строк на странице 108 (N). Например, метаданные 122 перераспределения имеют информационную запись 0, соответствующую строке 0, информационную запись 1, соответствующую строке 1, информационную запись 2, соответствующую строке 2 и т.д. Метаданные 122 перераспределения сконфигурированы для логической реорганизации строк в пределах страницы 108(N) путем переключения или смещения информационных записей таким образом, чтобы надежные или рабочие строки создавали логическое непрерывное пространство в пределах страницы 108(N), и ненадежные или вышедшие из строя строки логически перенаправлялись на участок, расположенный за пределом страницы 108 (N), где были также установлены метаданные 122 перераспределения. Величина пространства для хранения данных, которое требуется для каждой информационной записи в метаданных 122 перераспределения, может быть равна числу битов,

основывающемся на логарифме от числа строк на странице 108(N). Так как метаданные 122 перераспределения управляют и перенаправляют пространство хранения данных для страницы 108(N), метаданные 122 перераспределения можно сохранять с избыточностью в случае, когда некоторые из битов, где были сохранены метаданные 122 перераспределения, отказали или повреждены. Более того, модуль 118 кластеризации можно выполнить с возможностью восстановления метаданных 122 перераспределения, если метаданные 122 перераспределения повреждены в результате тестирования аппаратных средств с помощью операций записи.

Как показано на этапе 200(D), модуль 118 кластеризации переключает информационную запись для строки 1 (например, фактический отказ на стадии 200(C)) на информационную запись для строки 6 в нижней части страницы. Таким образом, согласно метаданным 122 перераспределения с переключенными информационными записями, строка 6 становится логической вышедшей из строя строкой, и строка 1 становится логической рабочей строкой. Контроллер 104 памяти сохраняет данные в фактической строке 6, которая была логически переключена на строку 1, с тем чтобы создать непрерывное пространство для хранения данных.

Далее, при выходе из строя фактической строки 3, модуль 118 кластеризации переключает информационную запись для вышедшей из строя строки 3 на информационную запись для рабочей строки 5, которая представляет собой следующую рабочую строку, доступную для переключения на нее во время процесса перераспределения. Опять же, путем переключения информационных записей модуль 118 кластеризации создает логический участок непрерывного пространства памяти, который не является вызванным или разделенным вышедшими из строя строками памяти.

В различных вариантах осуществления каждую информационную запись в метаданных 122 перераспределения можно индексировать для отражения сдвига физического адреса (например, физическое местоположение первоначальной строки, связанной с информационной записью) и логического сдвига строки (например, физическое местоположение перераспределенной строки в случае, когда первоначальная строка вышла из строя). Таким образом, в различных вариантах осуществления метаданные 122 перераспределения обеспечивают преобразование адреса или местоположения путем указания того, что вышедшая из строя строка 1 была логически переключена на рабочую строку 6, и вышедшая из строя строка 3 была логически переключена на рабочую строку 5, как показано в виде реорганизации (например, переупорядочивания) информационных записей метаданных 122 перераспределения на стадии 200(D). Так как метаданные 122 перераспределения были созданы для и установлены на строке 7, информационная запись для строки 7 может быть пустой (например, "**") и не включать в себя какой-либо перенаправляющей информационной записи. В различных вариантах осуществления информационная запись для строки 7 может хранить избыточные метаданные перераспределения, как упомянуто выше.

Таким образом, исходные информационные записи в метаданных 122 перераспределения, соответствующие рабочим строкам на странице 108(N), на которые не были логически переключены вышедшие из строя строки, остаются такими же. Например, на стадии 200(D) информационные записи для строк 0, 2 и 4 в метаданных 122 перераспределения не были модифицированы или изменены, потому что они не поменялись местами с информационными записями для вышедшей из строя строки.

В различных вариантах осуществления метаданные 122 перераспределения используют указатель 202 границы, с тем чтобы модуль 118 кластеризации мог

идентифицировать следующую доступную рабочую строку, чтобы логически поменять ее местами с вышедшей из строя строкой. Поскольку модуль 118 кластеризации логически объединяет вместе вышедшие из строя строки на участке, расположенном за пределами страницы 108(N), указатель 202 границы указывает на следующую рабочую строку, которая отделяет логическое непрерывное пространство от логического вышедшего из строя пространства. Например, на стадии 200(D) указатель 202 границы указывает на фактическую строку 4 как на следующую рабочую строку, которая ограничивает и отделяет логические отказы (например, строки 5, 6 и 7), объединенные в нижней части страницы 108(N). Соответственно, когда модуль 116 обнаружения отказов обнаруживает другой отказ строки, модуль 118 кластеризации реорганизует страницу 108(N) памяти путем логического перенаправления, в метаданных 122 перераспределения, адреса или индекса вышедшей из строя строки на строку, идентифицируемую указателем 202 границы. Модуль 118 кластеризации может затем продвигать указатель 202 границы на следующую рабочую строку и обновлять метаданные 122 перераспределения.

В различных вариантах осуществления каждая строка на странице 108(N) выполнена с индикатором, который дает возможность модулю 114 выделения памяти, во время начального доступа, идентифицировать, была ли логически перераспределена эта строка в процессе кластеризации. Например, индикатор может представлять собой один бит, зарезервированный на участке исправления ошибок, показывающий "да" в том случае, когда строка вышла из строя и была перераспределена, или "нет", когда строка не вышла из строя и данные, к которым осуществляется доступ, хранятся в строке. Когда строка вышла из строя, модуль 118 кластеризации может установить этот индикатор для указания того, что строка была перенаправлена или перераспределена. По меньшей мере в одном варианте осуществления модуль 114 выделения памяти и/или контроллер 104 памяти может определить местоположение метаданных 122 распределения, так как модуль 114 выделения памяти и/или контроллер 104 памяти знают, что метаданные 122 перераспределения можно расположить на первой строке в верхней части страницы 108(N) или на последней строке в нижней части страницы 108(N). В других вариантах осуществления индикатор для каждой строки можно дополнительно выполнить таким образом, чтобы он имел дополнительную возможность указывать метаданные 122 перераспределения в пределах страницы 108(N).

Исходя из различий между стадией 200(B) и стадией 200(D) видно, что процессы кластеризации и перераспределения, обсужденные в данном документе, логически создают большее количество непрерывного пространства для более крупного выделения объектам. Например, если бы модуль 114 выделения памяти должен был выделить память в окрестности вышедших из строя строк 1 и 3 на стадии 200(B), модуль 114 выделения памяти мог бы сохранить данные в 1) строке 0, 2) строке 2 или 3) строках 4-7 (например, четыре непрерывных строки). После выполнения процессов кластеризации и перераспределения модуль 114 выделения памяти может сохранить данные в логически рабочих строках 0-4 (например, пять непрерывных строк). Как упомянуто выше, страница 108(N) на Фиг. 2 включает в себя восемь строк в иллюстративных целях и для облегчения понимания. Соответственно, выгода и преимущества процессов кластеризации и перераспределения могут быть более очевидными за счет использования примеров, где имеется больше строк (например, 64 строки на странице), больше вышедших из строя строк, рассредоточенных в пределах страницы, и т.д.

Фиг. 3 иллюстрирует две соседние страницы 2N и 2N+1 300, которые хранятся в модуле 106 памяти. Для дальнейшей максимизации величины непрерывного

пространства памяти, модуль 118 кластеризации можно выполнить с возможностью кластеризации вышедших из строя строк и метаданных 122 перераспределения на верхней части четных страниц (например, страниц $2N$) в модуле 106 памяти и в нижней части нечетных страниц (например, страниц $2N+1$) в модуле 106 памяти или наоборот.

5 Таким образом, количество непрерывного пространства памяти может охватывать две последовательные страницы - нижнюю часть страницы $2N$ и верхнюю часть страницы $2N+1$, как иллюстрировано на Фиг. 3.

В различных вариантах осуществления модуль 118 кластеризации может выполнить положительное или отрицательное приращение указателя 204 границы и организовать

10 информационные записи в метаданных 122 перераспределения для каждой страницы $108(N)$ в зависимости от того, происходит ли кластеризация в верхней части страницы (например, метаданные 122 перераспределения сохраняются в строке 0) или в нижней части страницы (например, метаданные 122 перераспределения сохраняются в последней строке страницы, то есть строке 7 в примере по Фиг. 2).

15 Примеры, приведенные выше, в отношении Фиг. 2 и 3, описывают области перераспределения памяти в виде отдельных страниц, которые соответствуют областям системы памяти (например, страницы $108(1) \dots 108(N)$). Однако другие реализации процессов перераспределения и кластеризации памяти могут не использовать области перераспределения памяти, которые имеют такую же степень детализации, как и области

20 системы памяти. Например, модуль 118 кластеризации может реализовать область перераспределения памяти, которая кратна числу областей памяти, используемых в модуле 106 аппаратной памяти. Таким образом, модуль 118 кластеризации может разместить метаданные 122 перераспределения, например, в верхней части или в нижней части каждой второй, четвертой, восьмой страницы. Более того, метаданные 122

25 перераспределения и указатель 102 границы можно сконфигурировать для ссылки на конкретную страницу, а также строку, которая была переключена. Это позволяет дополнительно устранить фрагментацию и создать более крупные участки непрерывной памяти.

По меньшей мере в одном варианте осуществления модуль 118 кластеризации можно

30 выполнить с возможностью перехода с одного размера области перераспределения памяти (например, с одной страницы) на другой размер области перераспределения памяти (например, на восемь страниц). Например, модуль 118 кластеризации может изначально кластеризовать вышедшие из строя строки в соответствии с детализацией на уровне отдельной страницы, но после того, как каждая группа страниц будет иметь

35 вышедшие из строя строки, модуль 118 кластеризации может, вместо этого, кластеризовать вышедшие из строя строки в соответствии с детализацией на уровне восьми страниц (например, область перераспределения памяти восьми страниц) устранить большую фрагментацию. В данном случае, системное программное обеспечение 102 и контроллер 104 памяти позволяют затем, соответственно,

40 реконфигурировать систему и обновить метаданные 122 перераспределения, чтобы отразить переход с одного размера области перераспределения памяти на другой размер области перераспределения памяти.

Иллюстративные процессы

Примерные операции описаны ниже со ссылкой на Фиг. 4 и 5. Процессы,

45 иллюстрированные в виде блок-схемы логической последовательности операций, которая изображает последовательность операций, которую можно реализовать в виде аппаратных средств, программного обеспечения или их комбинации. В контексте программного обеспечения операции представляют собой машиноисполняемые

инструкции, которые при их исполнении одним или более процессорами выполняют перечисленные операции. В общем машиноисполняемые инструкции включают в себя подпрограммы, программы, объекты, компоненты, модули, структуры данных и т.п., которые выполняют конкретные функции или реализуют конкретные абстрактные типы данных. Порядок, в котором описаны эти операции, не следует истолковывать как ограничивающий, и любое количество описанных операций можно объединить в любом порядке и/или параллельно для реализации процесса.

Фиг. 4 иллюстрирует примерный процесс 400, направленный на создание и/или обновление метаданных 122 перераспределения для отдельной страницы 108(N) в модуле 106 памяти. Например, примерный процесс 400 может быть связан с операцией записи, которая выполняется с помощью модуля 114 выделения памяти, расположенного в системном программном обеспечении 102, контроллере 104 памяти или их комбинации. Более того, контроллер 104 памяти может быть расположен в модуле 106 памяти или отдельно от модуля 106 памяти на стороне обработки системы наряду с системным программным обеспечением 102.

На этапе 402 контроллер 104 памяти принимает выданную инструкцию для записи, и контроллер 104 памяти пытается записать данные в строку на странице 108(N). Например, контроллер 104 памяти может принимать выданную инструкцию, когда иерархия кэш-памяти хочет исключить «грязную» (dirty) строку и поэтому запрашивает обратную запись. Контроллер 104 памяти можно затем выполнить с возможностью идентификации страницы 108(N) и строки для обратной записи.

На этапе 404 модуль 116 обнаружения отказов обнаруживает, что строка представляет собой вышедшую из строя строку. Например, модуль 116 обнаружения отказов обнаруживает и подтверждает, что строка является вышедшей из строя строкой и не является больше надежной для целей хранения, когда контроллер 104 памяти фактически записывает данные в строку и также временно сохраняет другую версию данных в буфере. Когда контроллер 104 памяти считывает данные обратно из строки, модуль 116 обнаружения отказов сравнивает считанные данные с другой версией данных, временно хранящихся в буфере. Если данные, считанные обратно из строки, не являются такими, как данные, временно хранящиеся в буфере, то модуль 116 обнаружения отказов определяет, что строка представляет собой вышедшую из строя строку.

На этапе 406 контроллер 104 памяти может передать отказ строки и данные, которые нельзя записать, в системное программное обеспечение 102. Например, контроллер 104 памяти может использовать буфер неудачных операций записи и сгенерированное прерывание для передачи информации об отказе в операционную систему. Контроллер 104 памяти передает эту информацию, так как вышедшие из строя строки на странице 108(N) могут содержать один или более объектов, которые используются компонентом управляемой среды исполнения или другим системным компонентом.

На этапе 408 модуль 114 выделения памяти системного программного обеспечения 102 может затем перевыделить или переместить, например, для управляемой среды выполнения, один или более объектов на странице 108(N) в другие участки памяти в модуле 106 памяти (например, на другие страницы). Это перевыделение освобождает всю страницу 108(N). В различных вариантах осуществления модуль 114 выделения памяти может затем запустить процесс перераспределения и инструктировать контроллер 104 памяти реорганизовать страницу 108(N) с помощью логической кластеризации вышедших из строя строк на участке, расположенном за пределами страницы 108(N).

Соответственно, на этапе 410 модуль 118 кластеризации контроллера 104 памяти

создает и обновляет метаданные 122 перераспределения для страницы 108(N) и выполняет кластеризацию вышедших из строя строк на участке, расположенном за пределами страницы 108(N), как обсуждено выше в отношении Фиг. 2 и 3. Например, модуль 118 кластеризации может определить, является ли вышедшая из строя строка

5 первой вышедшей из строя строкой для отдельной страницы 108(N). Если модулем 118 кластеризации определено, что вышедшая из строя строка является первой вышедшей из строя строкой, модуль 118 кластеризации создает метаданные 122 перераспределения для страницы 108(N) в соответствии с обсуждением, приведенным выше в отношении Фиг. 2 и 3. Соответственно, модуль 118 кластеризации может создать информационные

10 записи для каждой строки на странице 108(N), реализовать "ложный" отказ на верхней части или в нижней части страницы 108(N), с тем чтобы метаданные 122 перераспределения можно было сохранить в пределах страницы 108(N), переключить информационную запись для вышедшей из строя строки на информационную запись для рабочей строки на верхней части или на нижней части страницы, следующей за

15 метаданными 122 перераспределения, и установить индикатор на вышедшей из строя строке, чтобы указать, что вышедшая из строя строка была перераспределена. Таким образом, модуль 118 кластеризации начинает выполнять логическую кластеризацию вышедших из строя строк на участке, расположенном за пределами страницы (например, в верхней части или нижней части в зависимости от того, является ли страница четной

20 или нечетной страницей в модуле 106 памяти), тем самым создавая большее количество непрерывного пространства на странице 108(N).

В различных вариантах осуществления, если модулем 118 кластеризации определено, что вышедшая из строя строка не является первой вышедшей из строя строкой, то модуль 118 кластеризации обращается к указателю 202 границы для определения

25 следующей доступной рабочей строки, которая отделяет вышедший из строя кластер строк от логического непрерывного пространства памяти. Модуль 118 кластеризации может затем переключить информационную запись для последующей (например, после первой вышедшей из строя строки и "ложной" вышедшей из строя строки) вышедшей из строя строки на информационную запись для следующей доступной рабочей строки,

30 идентифицируемой указателем 202 границы, и установить индикатор на последующей вышедшей из строя строке для указания того, что вышедшая из строя строка была перераспределена.

В различных вариантах осуществления, например, процесс 400 может повторяться каждый раз, когда выходит из строя строка в пределах отдельной страницы 108(N).

35 Более того, в различных вариантах осуществления этот примерный процесс 400 выполняется для каждой из многочисленных различных страниц 108(1)...108(N), которые хранятся в модуле 106 памяти.

Фиг. 5 иллюстрирует примерный процесс 500, направленный на использование метаданных 122 перераспределения во время операции считывания памяти. Например,

40 примерный процесс 500 может быть связан с операцией считывания, выдаваемой одним из процессорных ядер 110(1)...110(M) (или средством предварительной выборки), которая пропускает иерархию кэш-памяти и поэтому передается в контроллер 104 памяти, который будет производить считывание из модуля 116 памяти. Контроллер 104 памяти может быть расположен в модуле 106 памяти или отдельно от модуля 106

45 памяти на стороне обработки системы наряду с системным программным обеспечением 102.

На этапе 502 контроллер 104 памяти пытается считывать данные из строки на странице 108(N). Например, контроллер 104 памяти может принимать выданный из

одного из процессорных ядер 110(1)...110(M) запрос доступа к объекту или части объекта, который хранится в модуле 106 памяти.

На этапе 504 контроллер 104 памяти идентифицирует строку в качестве перераспределенной строки (например, вышедшей из строя строки). По меньшей мере в одном варианте осуществления контроллер 104 памяти может определить, что строка представляет собой перераспределенную строку, путем обращения с использованием физического адреса к фактической строке, и проверить индикатор (например, один бит, зарезервированный на участке исправления ошибок), чтобы определить, что строка изношена и вышла из строя и что она была перераспределена.

В другом варианте осуществления контроллер 104 памяти хранит метаданные 122 перераспределения для одной или более страниц 108(1)...108(N) локальным образом в кэше 120 информации об отказах (например, метаданные 122 перераспределения для страниц, к которым в текущий момент времени осуществляется доступ страниц, к которым недавно осуществлялся доступ, и/или страниц, к которым предположительно будет осуществляться доступ в будущем), и контроллер памяти и/или модуль 114 выделения памяти может проверить метаданные 122 перераспределения, которые хранятся в кэше 120 информации об отказах, чтобы определить, была ли перераспределена строка, и определить перенаправление адреса.

На этапе 506 контроллер 104 памяти обращается к метаданным 122 перераспределения и использует метаданные 122 перераспределения, чтобы определить перенаправленную новую строку, которая была локальным образом переключена с вышедшей из строя строки и которая фактически хранит данные или часть объекта (например, строки 5 и 6 на Фиг. 2).

На этапе 508 контроллер 104 памяти считывает данные из новой строки.

Примерный процесс 500 по Фиг. 5 можно также повторять каждый раз, когда строка в пределах отдельной страницы 108(N) связана с доступом для чтения или запросом на чтение.

В различных вариантах, где метаданные 122 перераспределения кэшируются в кэше 120 информации об отказах, который может поддерживаться в контроллере 106 памяти, системная архитектура может иметь возможность сохранения времени и энергии при выполнении доступа для чтения. Например, для того чтобы контроллер 104 памяти располагал и считывал данные из новой строки, когда метаданные 122 перераспределения не хранятся в кэше 120 информации об отказах, контроллер 104 памяти может понадобиться осуществить доступ к модулю 106 памяти в трех различных случаях: (1) первый доступ, которым идентифицируется, что первоначальная строка была перераспределена, (2) второй доступ к местоположению метаданных 122 перераспределения в верхней или нижней части страницы и (3) третий доступ к перенаправленной строке, на которую указывают метаданные 122 перераспределения. Это позволяет увеличить время задержки на считывание для системы. Напротив, когда метаданные 122 перераспределения кэшируются в кэше 120 информации об отказах, контроллер 104 памяти может определить локальным образом перенаправление и затем выполнить один доступ к модулю 106 памяти для считывания данных из новой строки.

Примерные процессы, иллюстрированные на Фиг. 4 и Фиг. 5, можно реализовать с помощью системного программного обеспечения 102, контроллера 104 памяти или их комбинации совместно с управляемой средой исполнения. Более того, в дополнение к выполнению процесса кластеризации и перераспределения, которые уменьшают или устраняют фрагментацию, как обсуждено выше, системное программное обеспечение

102 и/или контроллер 104 памяти можно также выполнить с возможностью отслеживания отказов строк на отдельных страницах 108(1)...108(N) и сохранять карту отказов (например, в кэше 120 информации об отказах). В различных вариантах осуществления отказы строк могут сообщаться в системы со средой исполнения во избежание отказов строк при выделении памяти.

В некоторых вариантах осуществления процессы кластеризации и перераспределения, обсужденные в данном документе, могут работать совместно с другими технологиями обработки отказов, выполняемыми для систем памяти. Например, контроллер 104 памяти может осуществлять доступ к метаданным 122 перераспределения для одной или более страниц 108(1)...108(N) после перераспределения и кластеризации вышедших из строя строк в пределах отдельных страниц. Так как, по меньшей мере, некоторые страницы 108(1)...108(N) не являются больше идеальными, а скорее являются несовершенными, так как они имеют вышедшие из строя строки, система программного обеспечения 102 или контроллер 104 памяти может объединить или разделить страницы, основываясь на том, являются ли страницы идеальными или несовершенными, и количество непрерывного пространства, которое может быть обеспечено несовершенной страницей, может быть сообщено, например, с управляемой средой исполнения. В другом примере, контроллер 104 памяти может перевыделить данные, которые хранятся на странице 108(1)...108(N), в случае когда строка выходит из строя.

В различных вариантах осуществления процессы перераспределения и кластеризации, описанные в данном документе, можно также использовать в процессе сбора ненужных данных (мусора), который высвобождает строки памяти и перемещает данные и объекты в пределах модуля 106 памяти. Например, средство сбора ненужных данных может использовать карту отказов, полученную из системного программного обеспечения 102 при выделении памяти. В результате, средство сбора ненужных данных позволяет избежать вышедших из строя строк при выделении памяти в процессе сбора ненужных данных.

Хотя страницы 108(1)...108(N) в модуле 106 памяти, обсужденные в данном документе, можно описать как РСМ, следует понимать, что технологии, описанные в данном документе, также применимы и к другим типам изнашиваемой памяти и что области памяти и сегменты памяти можно изменять, основываясь на системной архитектуре (например, сегменты могут иметь большую или меньшую степень детализации по сравнению со строками, и области могут иметь большую или меньшую детализацию по сравнению со страницами). Более того, модуль 106 памяти и любая другая память, которая может хранить системное программное обеспечение 102 и/или контроллер 104 памяти, могут также содержать машиночитаемые носители информации, включающие в себя по меньшей мере два типа машиночитаемых носителей информации, а именно компьютерные носители информации и коммуникационные носители информации.

Компьютерные носители информации включают в себя энергозависимые и энергонезависимые, съемные и несъемные носители, реализованные любым способом или с помощью любой технологии для хранения информации, такой как машиночитаемые инструкции, структуры данных, модули программ или другие данные. Компьютерные носители информации включают в себя, но не ограничиваются этим, оперативное запоминающее устройство (RAM), постоянное запоминающее устройство (ROM), электрически стираемое программируемое постоянное запоминающее устройство (EEPROM), флэш-память или любую другую технологию памяти, постоянное запоминающее устройство на компакт-диске (CD-ROM), цифровые универсальные диски (DVD) или другое оптическое запоминающее устройство, кассеты с магнитной

лентой, магнитную ленту, запоминающее устройство на основе магнитного диска или другие магнитные запоминающие устройства, или любую другую непередающую среду, которую можно использовать для хранения информации для доступа к вычислительному или серверному устройству.

5 Напротив, коммуникационные носители информации позволяют реализовать машиночитаемые инструкции, структуры данных, модули программ или другие данные в виде модулированного сигнала данных, такого как несущая волна или другой механизм передачи. Как определено в данном документе, компьютерные носители информации не исключают коммуникационные носители информации.

10 Заключение

Хотя изобретение было описано на языке, характерном для структурных признаков и/или методологических действий, должно быть понятно, что объем изобретения, определяемый прилагаемой формулой изобретения, необязательно ограничен конкретными признаками или операциями, описанными выше. Скорее, конкретные
15 признаки и операции, описанные выше, приведены в виде примеров для реализации пунктов формулы изобретения.

(57) Формула изобретения

1. Система для обработки сегментов памяти, вышедших из строя, содержащая:
20 аппаратный модуль памяти, включающий в себя множество областей памяти, причем отдельные области памяти включают в себя множество сегментов;

модуль обнаружения отказов, который обнаруживает вышедший из строя сегмент в области памяти из данного множества областей памяти;

25 модуль кластеризации, который логически кластеризует вышедший из строя сегмент посредством перераспределения вышедшего из строя сегмента в рабочий сегмент, расположенный на краю упомянутой области памяти, причем рабочий сегмент выбирается для максимизации непрерывного пространства памяти в пределах данной области памяти.

2. Система по п.1, в которой модуль кластеризации дополнительно создает
30 метаданные перераспределения для области памяти и сохраняет метаданные перераспределения для области памяти в пределах области памяти, причем метаданные перераспределения включают в себя информационную запись для каждого из множества сегментов в области памяти.

3. Система по п.2, в которой модуль кластеризации сохраняет метаданные
35 перераспределения в первом сегменте верхней части области памяти или в последнем сегменте в нижней части области памяти.

4. Система по п.2, в которой модуль кластеризации переключает, в метаданных перераспределения, первую информационную запись, связанную с вышедшим из строя сегментом, на вторую информационную запись, связанную с рабочим сегментом.

40 5. Система по п.1, в которой модуль кластеризации дополнительно создает метаданные перераспределения для области перераспределения памяти, которая включает в себя упомянутую область памяти и дополнительную область памяти из упомянутого множества областей памяти и сохраняет метаданные перераспределения для области перераспределения памяти в верхней части упомянутой области памяти
45 или в нижней части упомянутой области памяти, причем метаданные перераспределения включают в себя информационную запись для каждого из сегментов в упомянутых области памяти и дополнительной области памяти, и упомянутое непрерывное пространство памяти охватывает эти область памяти и дополнительную область памяти.

6. Система по п.1, при этом рабочий сегмент, расположенный на краю области памяти, представляет собой по меньшей мере одно из самого верхнего имеющегося в наличии рабочего сегмента области памяти и самого нижнего имеющегося в наличии рабочего сегмента области памяти.

5 7. Система по п.1, в которой аппаратный модуль памяти содержит память на основе фазового перехода.

8. Система по п.1, дополнительно содержащая модуль выделения памяти, который выделяет пространство хранения в упомянутом множестве сегментов памяти, при этом модуль выделения памяти представляет собой часть операционной системы.

10 9. Система по п.1, в которой по меньшей мере один из модуля обнаружения отказов и модуля кластеризации представляет собой часть контроллера памяти.

10. Система по п.9, в которой контроллер памяти размещен на аппаратном модуле памяти.

15 11. Система по п.1, в которой каждая из упомянутого множества областей памяти содержит страницу памяти и каждый сегмент содержит строку памяти.

12. Система по п.1, в которой модуль кластеризации использует указатель границы, чтобы идентифицировать рабочий сегмент как следующий логический рабочий сегмент, смежный с последним логическим вышедшим из строя сегментом, кластеризованным на упомянутый край области памяти.

20 13. Система по п.1, в которой модуль кластеризации создает большую величину пространства непрерывной памяти в пределах упомянутого множества областей памяти.

14. Способ обработки сегментов памяти, вышедших из строя, содержащий этапы, на которых:

25 принимают запрос доступа к данным, хранящимся в сегменте, который является частью области памяти, поддерживаемой в модуле памяти;

идентифицируют данный сегмент в качестве вышедшего из строя сегмента, который был перераспределен в рабочий сегмент, расположенный на краю упомянутой области памяти, чтобы максимизировать непрерывное пространство памяти в пределах данной области памяти;

30 обращаются к метаданным перераспределения, которые показывают местоположение на упомянутом краю области памяти для рабочего сегмента; и считывают данные из рабочего сегмента.

15. Способ по п.14, в котором:

35 метаданные перераспределения хранятся в кэше, локальном по отношению к контроллеру памяти;

при упомянутой идентификации сегмента в качестве вышедшего из строя сегмента проверяют метаданные перераспределения, хранящиеся в этом кэше, локальном по отношению к контроллеру памяти; и

40 упомянутое обращение к метаданным перераспределения, которые показывают упомянутое местоположение на краю области памяти для рабочего сегмента, выполняется локальным образом в контроллере памяти.

16. Способ по п.14, в котором метаданные перераспределения хранятся в верхней части упомянутой области памяти или в нижней части упомянутой области памяти.

45 17. Способ по п.14, в котором метаданные перераспределения кластеризуют вышедшие из строя строки в верхнюю часть упомянутой области памяти или в нижнюю часть упомянутой области памяти.

18. Способ обработки сегментов памяти, вышедших из строя, содержащий этапы, на которых:

принимают запрос записи данных в сегмент в области памяти аппаратного модуля памяти;

определяют, что этот сегмент является вышедшим из строя сегментом; и

5 создают метаданные перераспределения, которые логически переключают вышедший из строя сегмент на рабочий сегмент, расположенный на краю упомянутой области памяти, чтобы создать больше непрерывного пространства памяти в данной области памяти.

10 19. Способ по п.18, в котором метаданные перераспределения логически кластеризуют один или более вышедших из строя сегментов в верхнюю часть упомянутой области памяти или в нижнюю часть упомянутой области памяти.

20. Способ по п.18, в которой метаданные перераспределения включают в себя указатель границы, который идентифицирует следующий используемый сегмент, на который должен быть переключен следующий вышедший из строя сегмент.

15

20

25

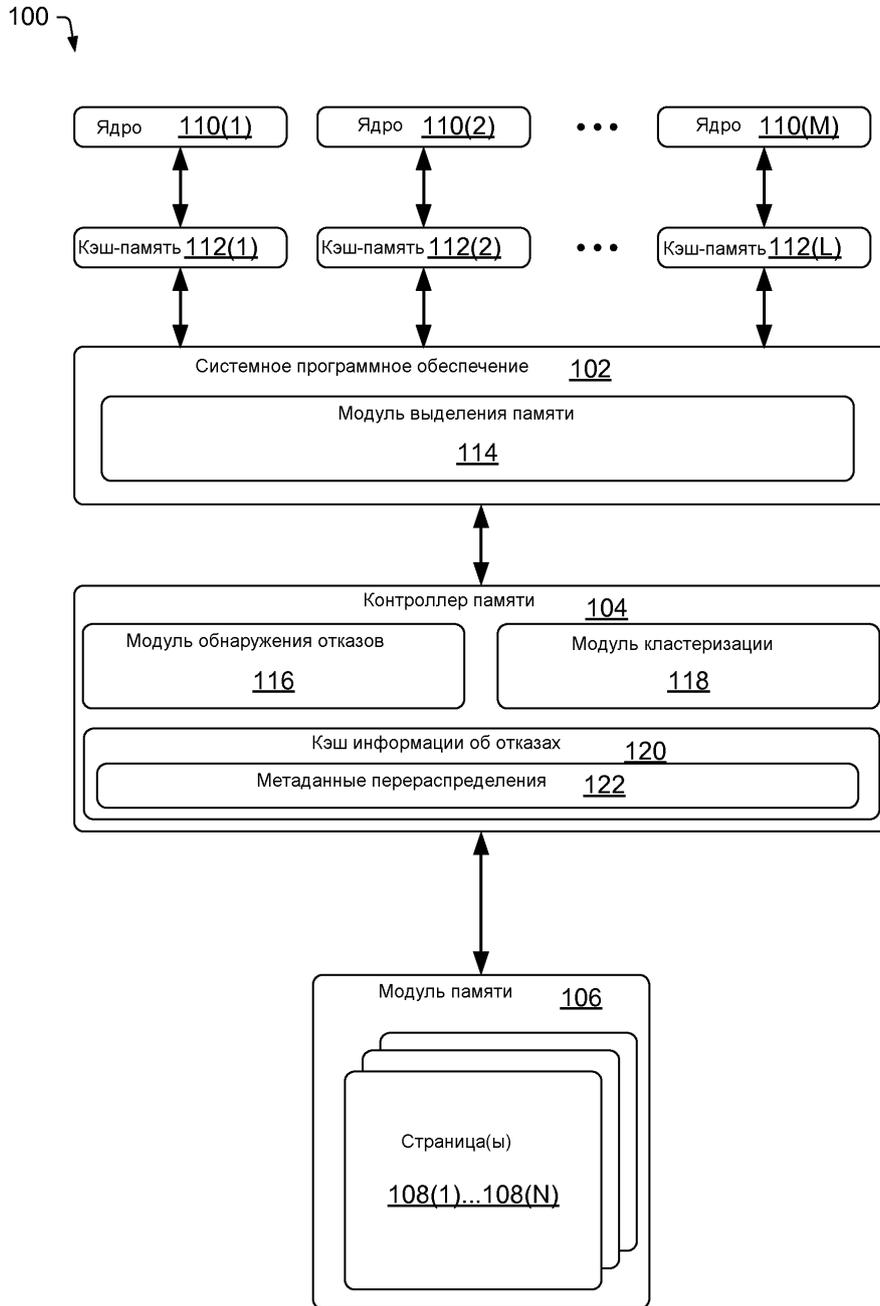
30

35

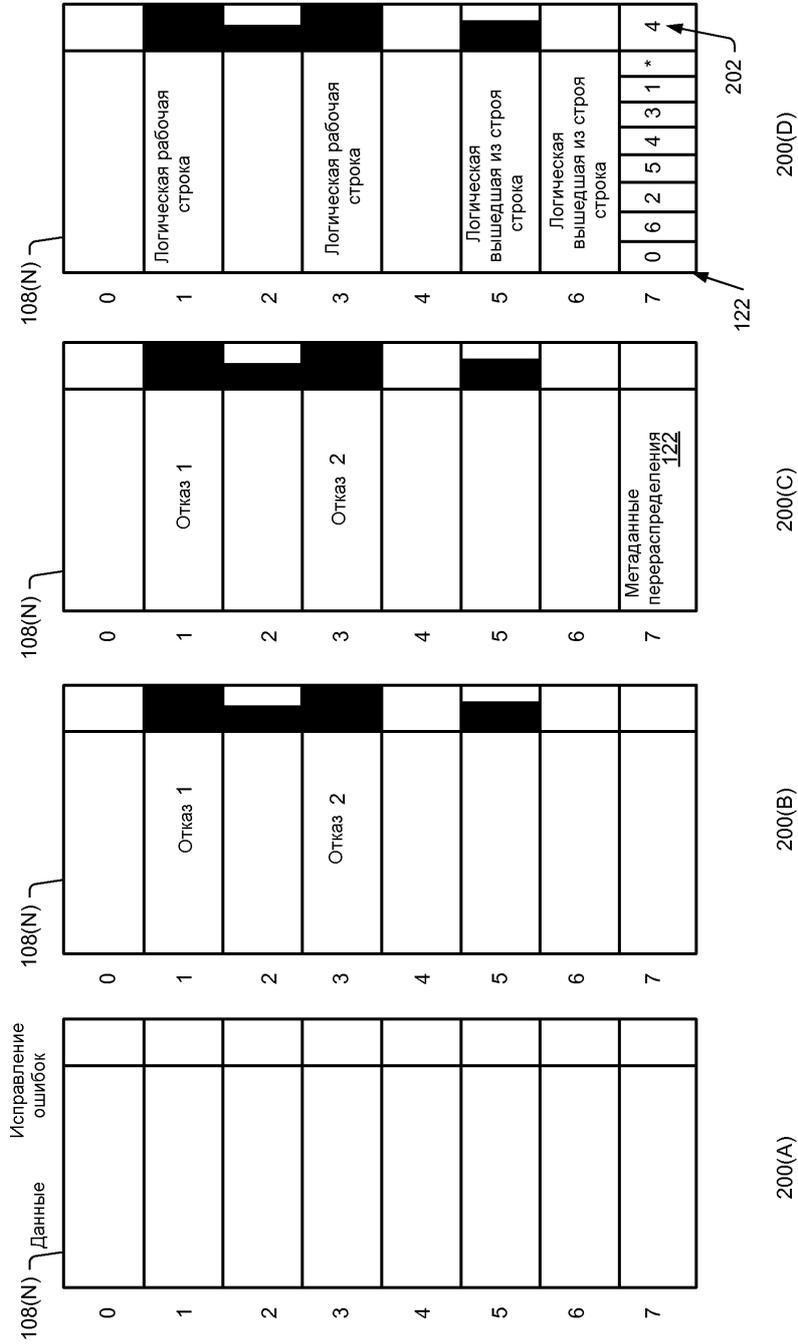
40

45

1/5



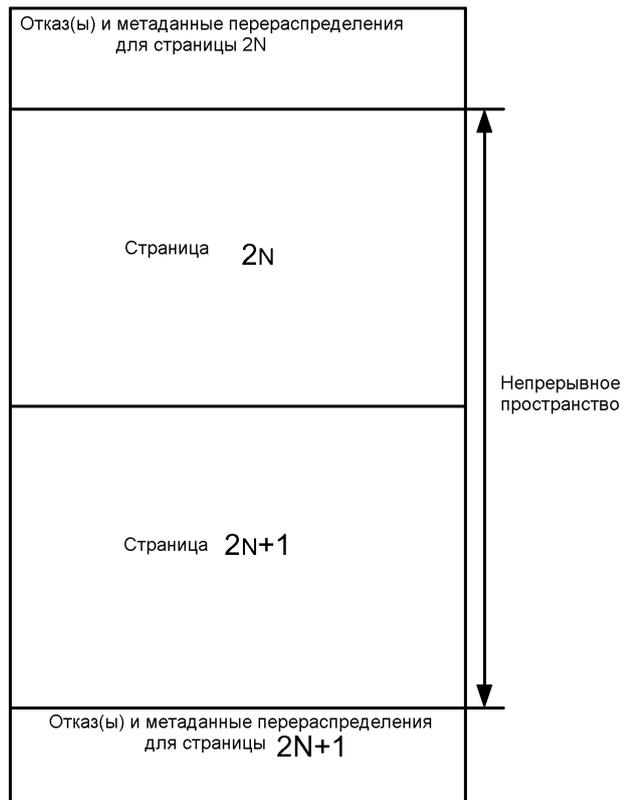
ФИГ.1



ФИГ. 2

3/5

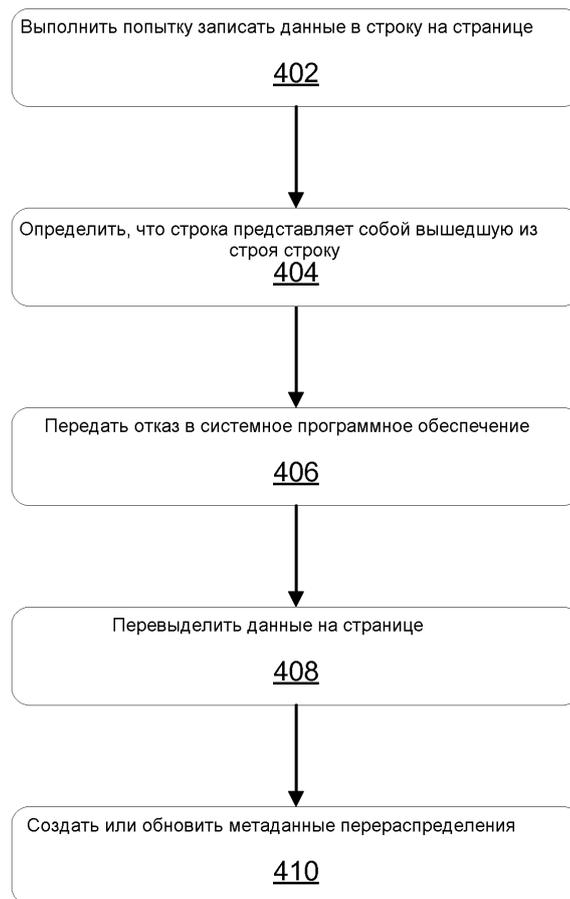
300 ↘



ФИГ.3

4/5

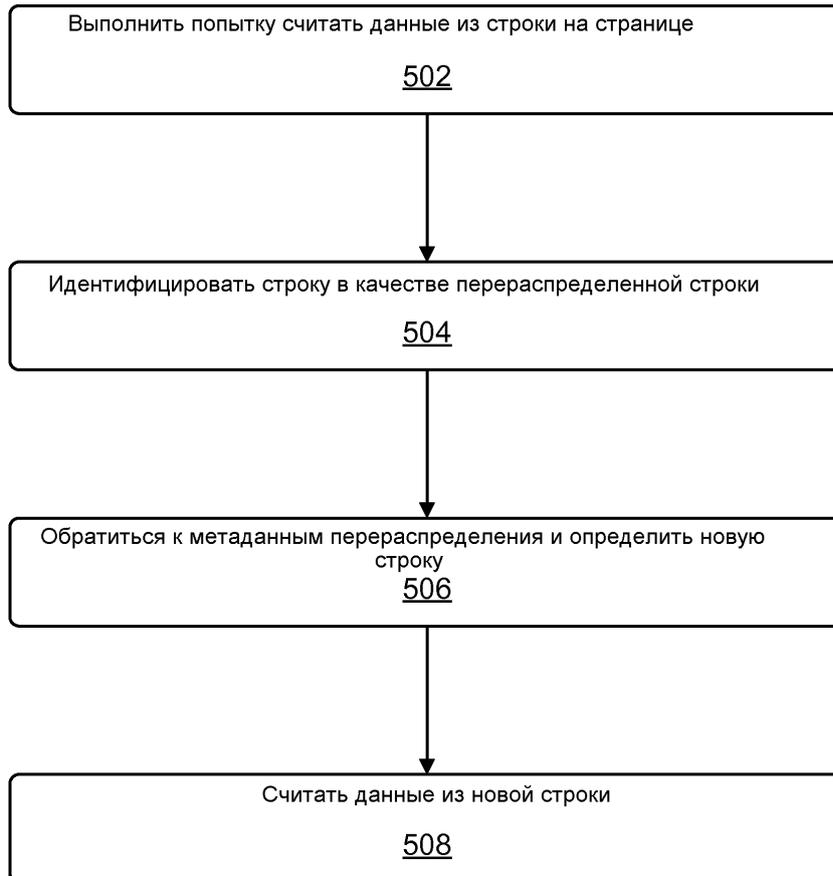
400 ↘



ФИГ.4

5/5

500 ↘



ФИГ.5