



**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2010139638/08, 16.02.2009**

(24) Дата начала отсчета срока действия патента:
16.02.2009

Приоритет(ы):

(30) Конвенционный приоритет:
28.03.2008 US 12/057,716

(43) Дата публикации заявки: **10.04.2012** Бюл. № 10

(45) Опубликовано: **10.01.2014** Бюл. № 1

(56) Список документов, цитированных в отчете о поиске: **US 6901522 B2, 31.05.2005. US 7093147 B2, 15.08.2006. US 7437581 B2, 14.10.2008. US 2005/0050373 A1, 03.03.2005. RU 2171490 C2, 27.07.2001.**

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: **27.09.2010**

(86) Заявка РСТ:
US 2009/034209 (16.02.2009)

(87) Публикация заявки РСТ:
WO 2009/120427 (01.10.2009)

Адрес для переписки:

**129090, Москва, ул. Б. Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и
Партнеры"**

(72) Автор(ы):

**МАРШАЛЛ Аллен (US),
ДЭН Иминь (US),
ДЖАДЖ Николас С. (US),
КИШАН Арун У. (US),
РИТЦ Эндрю Дж. (US)**

(73) Патентообладатель(и):

МАЙКРОСОФТ КОРПОРЕЙШН (US)

**(54) ЭНЕРГОСБЕРЕГАЮЩЕЕ ПЛАНИРОВАНИЕ ПОТОКОВ И ДИНАМИЧЕСКОЕ
ИСПОЛЬЗОВАНИЕ ПРОЦЕССОРОВ**

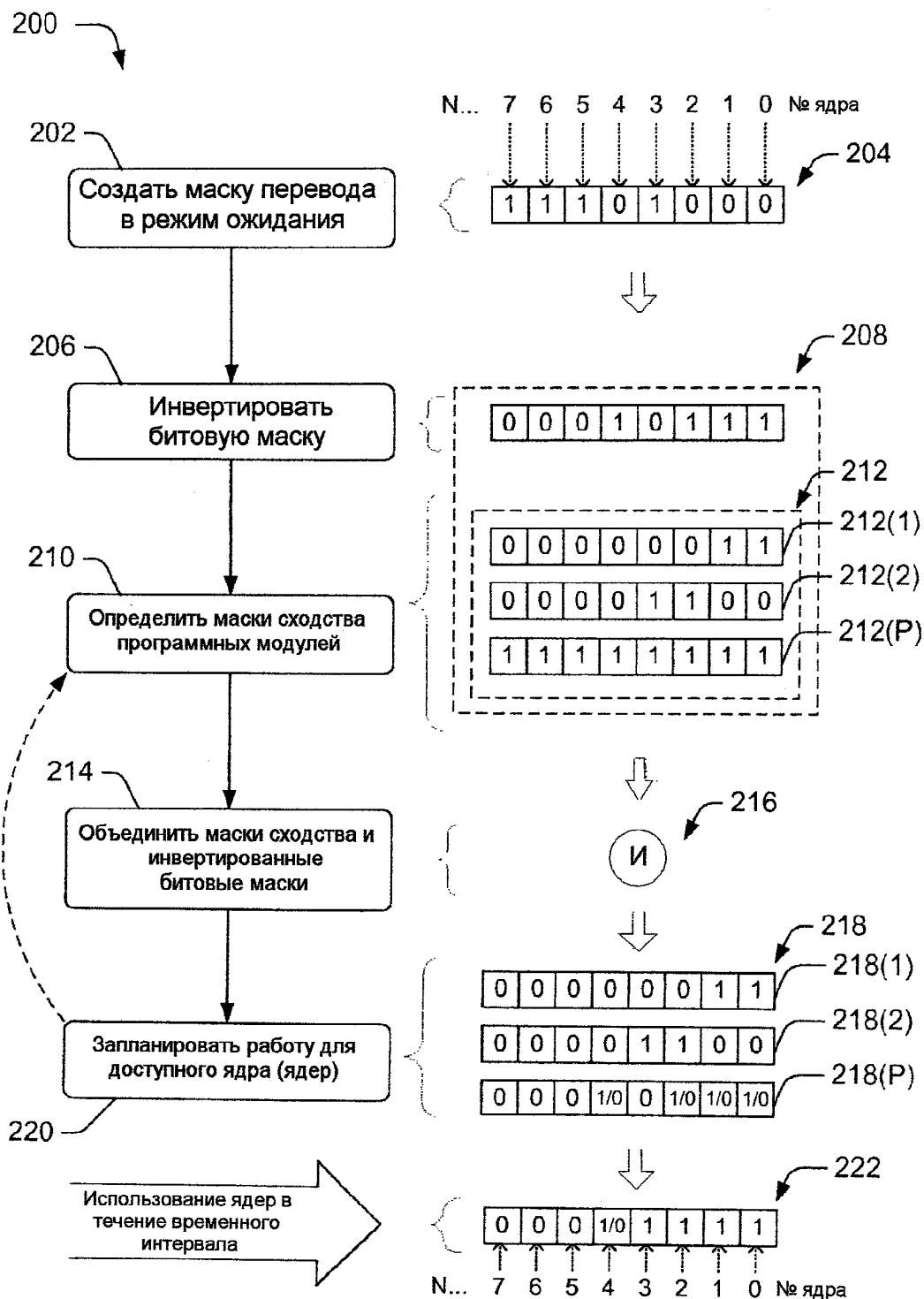
(57) Реферат:

Изобретение относится к средствам обеспечения энергосберегающего планирования потоков и динамического использования процессоров. Технический результат заключается в уменьшении потребления электроэнергии. Определяют, какие ядра из упомянутого множества ядер активно выполняют работу. Создают маску приостановки ядер с использованием битового значения для представления приостановленного или работающего

состояния ядра. Определяют маски привязки потока к процессору, представляющие одно или более ядер, которым назначена обработка потока. Обеспечивают, по меньшей мере, часть плана производительности и энергосбережения для ядер путем объединения инвертированной маски приостановки ядер и масок привязки потока к процессору, используя оператор "И", чтобы создать набор доступных процессоров. Вычисляют, какие ядра обозначаются как приостановленные или работающие, на основе, по меньшей мере частично, набора доступных

процессоров. Приостанавливают по меньшей мере одно из ядер, активно выполняющих работу, на основе, по меньшей мере частично, политики мощности, указывающей, что это по

меньшей мере одно из ядер, активно выполняющих работу, обозначено как приостановленное ядро. 3 н. и 15 з.п. ф-лы, 8 ил.



Фиг. 2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) ABSTRACT OF INVENTION(21)(22) Application: **2010139638/08, 16.02.2009**(24) Effective date for property rights:
16.02.2009

Priority:

(30) Convention priority:
28.03.2008 US 12/057,716(43) Application published: **10.04.2012 Bull. 10**(45) Date of publication: **10.01.2014 Bull. 1**(85) Commencement of national phase: **27.09.2010**(86) PCT application:
US 2009/034209 (16.02.2009)(87) PCT publication:
WO 2009/120427 (01.10.2009)

Mail address:

**129090, Moskva, ul. B. Spasskaja, 25, str.3, OOO
"Juridicheskaja firma Gorodisskij i Partnery"**

(72) Inventor(s):

**MARShALL Allen (US),
DEhN Imin' (US),
DZhADZh Nikolas S. (US),
KIShAN Arun U. (US),
RITTs Ehndrju Dzh. (US)**

(73) Proprietor(s):

MAJKROSOFT KORPOREJShN (US)**(54) POWER-SAVING STREAM SCHEDULING AND DYNAMIC USE OF PROCESSORS**

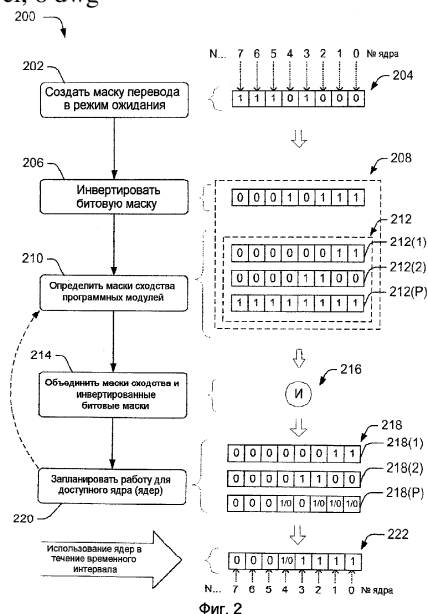
(57) Abstract:

FIELD: information technology.

SUBSTANCE: method involves determining which cores of multiple cores are actively working; creating a core parking mask using a bit value to represent a parked or unparked status of a core; determining stream processor affinity masks representing one or more cores on which a stream is assigned to be processed; providing at least a portion of the performance and power saving plan for the cores by combining an inversion of the core parking mask and the stream processor affinity masks using an "AND" operator to create an available processor set; calculating which cores are designated as parked or unparked based at least in part on the available processor set; and parking at least one of the cores actively working based at least in part on the power policy indicating that at least one of the cores actively working is designated as a parked core.

EFFECT: reduced power consumption.

18 cl, 8 dwg



Уровень техники

Управление энергопотреблением в компьютерной системе является важным для увеличения эксплуатационной возможности батареи и уменьшения общего энергопотребления, что может быть выгодным как с финансовой точки зрения, так и с экологической точки зрения. Даже для немобильных компьютеров уменьшение потребностей в электроэнергии полезно для сохранения важных глобальных ресурсов и продления работы при расчете на систему резервного батарейного питания, например, во время отключения энергоснабжения.

Хотя большинство компонентов вычислительной системы используют энергию во время работы системы, процессор использует непропорциональную долю энергии системы. Многие компьютерные системы, включая потребительские системы, включают в себя несколько процессоров и/или процессоры с несколькими ядрами. Несколько процессоров дают компьютерам возможность выполнять растущие уровни обработки параллельно, однако дополнительные процессоры также могут увеличивать энергопотребление. Большинство современных процессоров имеют состояния простоя с очень низкой мощностью, которые могут применяться к каждому ядру в многоядерной системе и которые могут управляться операционной системой. К тому же частота процессора может масштабироваться на основе ядра или группы ядер, чтобы уменьшить использование энергии системой.

Сущность изобретения

Это краткое изложение сущности изобретения предоставляется, чтобы представить упрощенные идеи обеспечения энергосберегающего планирования потоков и динамического использования процессоров, которое дополнительно описывается ниже в подробном описании. Это краткое изложение сущности изобретения не предназначено ни для определения существенных признаков заявленного изобретения, ни для использования в установлении объема заявленного изобретения.

Раскрываются типовые методики и устройства для обеспечения энергосберегающего планирования потоков и динамического использования процессоров. В соответствии с одним или несколькими вариантами осуществления, многоядерная система контролируется для определения активности ядер. Извлекается политика мощности для запуска плана производительности и энергосбережения для ядер. Одно или несколько ядер в многоядерной системе приостанавливаются (помещаются в заданное системой состояние малой мощности) на основе политики мощности и активности ядер. Когда одно или несколько ядер приостанавливаются ("паркуются"), работающие ядра остаются обрабатывать всю оставшуюся активность системы. В некоторых вариантах осуществления политика мощности может быть изменена для включения в нее дополнительных факторов, влияющих на энергосбережение или производительность системы. По меньшей мере в одном другом варианте осуществления многоядерная система может динамически регулировать состояние питания у одного или нескольких работающих ядер в дополнение к приостановке одного или нескольких ядер.

Краткое описание чертежей

Подробное описание изобретения описывается со ссылкой на прилагаемые чертежи. На фигурах крайняя левая цифра (цифры) в номере ссылки определяет фигуру, в которой номер ссылки появляется первый раз. Одинаковый номер ссылки на разных фигурах ссылается на аналогичные или идентичные элементы.

Фиг.1 - иллюстративная система, которая может использоваться для реализации по меньшей мере одного варианта осуществления энергосберегающего планирования

потоков и динамического использования процессоров.

Фиг.2 показывает блок-схему алгоритма по меньшей мере одного варианта осуществления процесса создания маски приостановки ядер и реализации маски с помощью планировщика потоков, чтобы разрешить распределение работы процессорам.

Фиг.3А и 3В показывают иллюстративные использования ядер в соответствии по меньшей мере с одним вариантом осуществления раскрытия изобретения. Точнее говоря, фиг.3В показывает пояснительную модификацию использований ядер в ответ на использование ядра в пояснительной системе, которое показано на фиг.3А.

Фиг.4 показывает блок-схему алгоритма по меньшей мере одного варианта осуществления пояснительного процесса оценивания планирования потоков и динамического использования процессоров и определения измененной конфигурации для процессоров.

Фиг.5 показывает блок-схему алгоритма по меньшей мере одного варианта осуществления пояснительного процесса обеспечения энергосберегающего планирования потоков и динамического использования процессоров.

Фиг.6 показывает другую блок-схему алгоритма по меньшей мере одного варианта осуществления пояснительного процесса обеспечения энергосберегающего планирования потоков и динамического использования процессоров, дополнительно включающего учет простоя доменов.

Фиг.7 показывает иллюстративную блок-схему алгоритма по меньшей мере одного варианта осуществления предоставления дополнительных настроек для энергосберегающего планирования потоков и динамического использования процессоров.

Подробное описание

Процессоры могут задействовать состояния простоя с низкой мощностью, включающие состояние простоя, которое не потребляет энергии (ноль ватт). Операционная система может вводить один или несколько процессоров (или просто "ядер") в состояние простоя (то есть состояние ожидания процессора), когда отсутствует полезная работа для выполнения. Максимизация времени, проведенного в этих состояниях малой мощности, может увеличить энергетическую эффективность системы и/или увеличить производительность батареи. В дополнение к этим состояниям простоя процессора процессоры также могут предоставлять средства управления для масштабирования частоты процессора либо отдельно, либо в сочетании с одновременным понижением напряжения на ядре процессора. Эти средства управления вместе могут называться функциями управления питанием процессора (PPM).

Процессоры могут способствовать выполнению миллиардов исполнений в секунду. Обладая такой высокой способностью для исполнения машинных команд, процессоры могут иметь значительное колебание в рабочей нагрузке за короткие периоды времени. Например, такие короткие периоды, как задержка между нажатиями клавиш машинистки, могут дать возможность PPM мгновенно уменьшить мощность процессора или даже войти в короткое состояние ожидания. Хотя доля секунды в мощности может показаться незначительной, за длительные периоды времени совокупное энергосбережение может быть значительным.

Поэтому PPM может снизить потребности в энергии путем перевода неиспользуемых процессоров в состояние малой мощности или состояние ожидания ("приостановленное" состояние), когда процессоры не имеют достаточной рабочей

нагрузки, чтобы обосновать состояния с большей мощностью. Приостановленные ядра могут помещаться в состояние простоя процессора (C-состояние в интерфейсе ACPI), использующего минимальное количество энергии или вообще никакой энергии. Активная работа, которая должна выполняться в системе, будет

уплотнена по времени на работающих процессорах. Соответственно, методики и устройства для содействия обеспечению энергосберегающего планирования потоков и динамического использования процессоров раскрываются в этом документе в различных разделах, которые следуют

Пояснительное окружение

Фиг.1 - пояснительная система 100, которая может использоваться для реализации по меньшей мере одного варианта осуществления энергосберегающего планирования потоков и динамического использования процессоров. Система 100 включает в себя вычислительное устройство 102. Например, вычислительное устройство в числе других возможных вычислительных устройств может быть мобильным компьютером 102(1), настольным компьютером 102(2) и/или сервером 102(N). В самой базовой конфигурации вычислительное устройство 102 обычно включает в себя один или несколько процессоров 104 ("процессоры"). Например, процессоры 104 могут быть по меньшей мере одним из нескольких независимых процессоров, сконфигурированных параллельно или последовательно, и многоядерным модулем обработки, отдельно или в различных сочетаниях. Многоядерный процессор может содержать два или более процессора ("ядра"), заключенных в одну микросхему или интегральную схему. Термины "процессор", "ядро" и "логический процессор" могут использоваться взаимозаменяемо во всем этом раскрытии изобретения, пока специально не указано иное со ссылкой на конкретный элемент.

К тому же вычислительное устройство 102 включает в себя системную память 106. В зависимости от точной конфигурации и типа вычислительного устройства системная память 106 может быть энергозависимой (например, RAM) энергонезависимой (например ROM, флэш-память и т.д.) или каким-нибудь сочетанием из двух типов. Системная память 106 обычно содержит в себе операционную систему 108, один или несколько программных модулей 110 и может включать в себя программные данные 112.

Операционная система 108 может включать в себя диспетчер 114 питания ядра, который несет ответственность за управление использованием функций управления питанием процессора (PPM). Диспетчер 114 питания ядра может регулировать производительность (например, скорость) процессоров 104, используя состояние производительности (p-состояние) или состояние линейной регулировки (t-состояние). Например, диспетчер 114 питания ядра может уравнивать энергопотребление процессоров 104 с текущей рабочей нагрузкой, чтобы сберегать энергию, когда это возможно. Дополнительно или в качестве альтернативы диспетчер 114 питания ядра может разрешить процессорам 104 обеспечивать максимальную производительность обработки в ответ на потребность в рабочей нагрузке. Дополнительно диспетчер 114 питания ядра может переводить один или несколько процессоров 104 в состояние ожидания с малой мощностью, когда никакие активные потоки не готовы к выполнению, например потоки программного модуля 110.

Операционная система 108 может предоставлять модули для организации очередей, планирования, приоритизации и диспетчеризации единиц работы (потоков) по всем доступным процессорам 104 в системе 100, которые могут быть представлены как

коллекция модулей, называемая вместе планировщиком 116 потоков ядра. Когда активный поток готов к выполнению, планировщик 116 ядра с помощью одного или нескольких модулей распределяет поток любому доступному ядру для обработки.

Вообще говоря, диспетчер 114 питания ядра и планировщик 116 ядра имеют конкурирующие интересы в управлении и работе процессоров 104. Диспетчер питания ядра конфигурируется для минимизации энергопотребления процессоров 104, и поэтому пытается снизить частоту и/или состояние питания у одного или нескольких процессоров 104. Наоборот, планировщик 116 ядра конфигурируется для максимизации производительности обработки и соответственно способствует распределению работы всем процессорам 104.

Вычислительное устройство 102 может обладать дополнительными признаками или функциональными возможностями. Например, вычислительное устройство 102 также может включать в себя дополнительные устройства хранения данных (съемные и/или несъемные), такие как, например, магнитные диски, оптические диски или лента. Такое дополнительное хранилище иллюстрируется на фиг.1 с помощью съемного запоминающего устройства 118 и несъемного запоминающего устройства 120.

Компьютерные носители информации могут включать в себя энергозависимые и энергонезависимые, съемные и несъемные носители, реализованные любым способом или технологией для хранения информации, такой как машиночитаемые команды, структуры данных, программные модули или другие данные. Системная память 106, съемное запоминающее устройство 118 и несъемное запоминающее устройство 120 и являются примерами компьютерных носителей информации. Таким образом, компьютерные носители информации включают в себя, но не ограничиваются, RAM, ROM, EEPROM, флэш-память или другую технологию памяти, компакт-диск, универсальные цифровые диски (DVD) или другое оптическое запоминающее устройство, магнитные кассеты, магнитную ленту, накопитель на магнитных дисках или другие магнитные запоминающие устройства или любой другой носитель, который может использоваться для хранения нужной информации и к которому можно обращаться с помощью вычислительного устройства 102. Любые такие компьютерные носители информации могут быть частью вычислительного устройства 102.

Вычислительное устройство 102 также может иметь одно или несколько устройств 122 ввода, таких как клавиатура, мышь, перо, устройство речевого ввода, устройство сенсорного ввода и т.д. Одно или несколько устройств 124 вывода, таких как дисплей, динамики, принтер и т.д., также могут включаться либо непосредственно, либо через соединение с вычислительным устройством 102.

Вычислительное устройство 100 также может включать в себя соединение 126 связи, которое позволяет устройству обмениваться данными с другими вычислительными устройствами, например по сети. Соединение 126 связи является одним примером средств связи. Средства связи, как правило, могут реализовываться с помощью машиночитаемых команд, структур данных или программных модулей. Термин "модулированный сигнал данных" означает сигнал, который имеет одну или несколько своих характеристик, установленных или измененных таким образом, чтобы кодировать информацию в сигнале. В качестве примера, а не ограничения, средства связи включают в себя проводные средства, такие как проводная сеть или прямое проводное соединение, и беспроводные средства, такие как акустические, радиочастотные, инфракрасные и другие беспроводные средства. Машиночитаемые носители могут быть любыми доступными носителями, к которым можно обращаться

с помощью вычислительного устройства 102. В качестве примера, а не ограничения, машиночитаемые носители могут содержать "компьютерные носители информации" и "средства связи".

Различные модули и методики могут быть описаны в этом документе в общем контексте исполняемых компьютером команд, например программных модулей, выполняемых одним или несколькими компьютерами или иными устройствами. Как правило, программные модули включают в себя процедуры, программы, объекты, компоненты, структуры данных и так далее для выполнения конкретных задач или реализации конкретных абстрактных типов данных. Эти программные модули и т.п. могут выполняться в виде машинного кода или могут загружаться и выполняться, например на виртуальной машине или в другой среде выполнения программы с оперативной компиляцией. Как правило, функциональные возможности программных модулей могут быть объединены или распределены по желанию в различных вариантах осуществления. Реализация этих модулей и методик может храниться или передаваться посредством какого-нибудь вида машиночитаемых носителей.

Конфигурация иллюстративного процессора

Фиг.2 показывает блок-схему алгоритма по меньшей мере одного варианта осуществления процесса 200 создания маски приостановки ядер и реализации маски с помощью планировщика потоков, чтобы разрешить распределение работы процессорам. Процесс 200 иллюстрируется в виде набора этапов в логической блок-схеме, которая представляет последовательность операций, которые могут быть реализованы в аппаратных средствах, программном обеспечении или их сочетании. Применительно к программному обеспечению этапы представляют собой исполняемые компьютером команды, которые при выполнении одним или несколькими процессорами выполняют перечисленные операции. Как правило, исполняемые компьютером команды могут включать в себя процедуры, программы, объекты, компоненты, структуры данных и т.п., которые выполняют конкретные функции или реализуют конкретные абстрактные типы данных. Порядок, в котором описываются операции, не предназначен для толкования в качестве ограничения, и любое количество описанных этапов может объединяться в любом порядке и/или параллельно для реализации процесса. Другие процессы, описанные во всем этом раскрытии изобретения в дополнение к процессу 200, следует интерпретировать соответствующим образом. Для целей обсуждения процесс 200 описывается со ссылкой на систему 100 из фиг.1.

Как показано на фиг.2, в системной памяти создается маска приостановки ядер на этапе 202. Например, диспетчер 114 питания ядра, который находится в операционной системе 108, может создать маску приостановки ядер на этапе 202. Пояснительная маска 204 приостановки ядер ("битовая маска" или просто "маска") может предоставлять ячейку, представляющую соответствующее ядро. Как показано на фиг.2, пояснительная система включает в себя восемь ядер, однако может использоваться больше или меньше ядер. Битовая маска 204 включает в себя двоичное значение в каждой ячейке, где "1" представляет приостановленное ядро, а "0" представляет работающее ядро. Приостановленное ядро является ядром, которое помещается в состояние ожидания с малой мощностью. В некоторых вариантах осуществления приостановленное ядро не имеет энергопотребления, соответственно использует ноль ватт. В некоторых вариантах осуществления ядра обладают зависимостями, например совместно используемой аппаратной схемой. Если оба ядра можно ввести в состояние малой мощности, зависимости также могут быть неявно

введены в состояние малой мощности. Таким образом, может выбираться маска приостановки ядер, которая будет максимизировать энергосбережение по сравнению с другими масками. Например, выключение всех ядер в однопроцессорном разьеме могло бы сэкономить больше энергии, чем выключение половины ядер в
 5 двухпроцессорных разьемах. Битовая маска 204 включает в себя четыре приостановленных ядра, пронумерованных (справа налево, от нуля до семи): 3, 5, 6 и 7. Из этого следует, что ядра 0, 1, 2 и 4 являются работающими ядрами.

В соответствии с одним или несколькими вариантами осуществления, битовую
 10 маску 204 можно инвертировать на этапе 206, чтобы создать инвертированную битовую маску 208. Например, диспетчер 114 питания ядра может создать инвертированную битовую маску 208. Инвертированная битовая маска включает в себя инвертированное двоичное значение для каждой ячейки (то есть ядра). Соответственно ядра, обозначенные "1", могут быть предпочтительными для
 15 обработки данных, тогда как ядра, обозначенные "0", не могут быть предпочтительными для обработки потоков приложений.

На этапе 210 определяется план приложений, например с помощью планировщика 116 ядра. Например, вычислительное устройство 102 может выполнять
 20 один или несколько программных модулей 110. Некоторые из программных модулей 110 могут включать в себя однопоточные программы, тогда как другие программные модули могут включать в себя многопоточные модули. Как правило, планировщик 116 ядра назначает каждый поток доступному ядру на основе некоторого количества факторов, таких как приоритет, доступность ядра, привязка
 25 (ограничения планирования) и другие факторы. Когда потоки численно превосходят доступные ядра, тогда планировщик 116 ядра чередует потоки или иным образом планирует потоки, чтобы обеспечивать, что потоки в конечном счете развиваются (то есть выполняются процессором). Процессоры 104 могут чередовать потоки на одном
 30 ядре много раз в секунду, соответственно предоставляя возможности планировщику 116 ядра эффективно назначать потоки доступным ядрам.

Хотя многие программные модули не распределяют потоки определенными процессорами, некоторые сложные программные модули могут требовать, чтобы поток выполнялся определенным процессором, что называется заданием привязки
 35 потока к процессору. Маски 212 привязки потоков программных модулей представляют собой ядра, запрошенные для обработки потоков, что определяется программными модулями 110. Например, первый программный модуль может иметь первую маску 212(1) привязки, которая указывает, что потоки должны планироваться
 40 планировщиком 116 ядра на ядра 0 и 1. Вторая маска 212(2) привязки, ассоциированная со вторым программным модулем, может указывать, что потоки могут назначаться на ядра 2 и 3, тогда как другой программный модуль может включать в себя маску 212(P) привязки, указывающую, что потоки могут назначаться на любое из доступных ядер (все ядра показаны выбранными). Следует отметить, что
 45 маска 212(P) привязки является особым случаем, в котором она практически не включает в себя ограничений для распределения потоков по ядрам.

В некоторых вариантах осуществления на этапе 214 маски 212 привязки программных модулей объединяются, по одной за раз, с инвертированной битовой
 50 маской 208, используя оператор 216 "И" для определения набора подходящих процессоров для набора 218 доступных процессоров. На этапе 220 первая маска 212(1) привязки используется для создания первого набора 218(1) доступных процессоров. Процесс 200 может включать в себя итеративный процесс из операций 210, 214 и 220

для каждого программного модуля (то есть для каждого сочетания инвертированной битовой маски 208 и маски 212 привязки). Таким образом, вторая маска 212(2) привязки используется для создания второго набора 218(2) доступных процессоров во время второй итерации операций 210, 214 и 220, и так далее.

Как обсуждалось выше, двоичные значения для каждого ядра (например, ядра 0, ..., ядра 7) используются для определения набора 218 доступных процессоров для планирования потоков. Оператор 216 "И" возвращает двоичное значение "1" ядра, где оба операнда (то есть объединенные двоичные значения для ядра) включают в себя "1", представляющую поток, связанный с определенным ядром. Например, когда первая маска 212(1) привязки объединяется с инвертированной битовой маской 208, оба ядра 0 и 1 являются активными ядрами и вернут двоичное значение ядра "1", тогда как оставшиеся ядра 2-7 включают в себя двоичное значение ядра "0", как пояснительно показано в первом наборе 218(1) доступных процессоров.

Вторая маска 212(2) привязки включает в себя значение ядра "1" у ядра 3, тогда как инвертированная битовая маска 208 указывает, что ядро 3 приостановлено. Планировщик 116 ядра может выбрать переопределение инвертированной битовой маски, чтобы обеспечить вторую маску 212(2) привязки, которая представлена во втором наборе 218(2) доступных процессоров, где ядро 3 включает в себя значение ядра "1", вследствие этого перенося работу на ядро, обозначенное как приостановленное в инвертированной битовой маске 208 (которое может быть работающим впоследствии). В некоторых вариантах осуществления поток может планироваться с использованием любого количества эвристических правил. Оптимальное ядро в привязке потока может использоваться наряду с игнорированием инвертированной битовой маски 208. Если оптимальное ядро приостановлено, то нейтрализация может включать в себя выбор процессора в том же узле NUMA (доступ к неоднородной памяти), что и предпочтительное ядро. Планировщик рассматривает инвертированную маску приостановки ядер в качестве рекомендации предпочтительных расположений для выполнения потока, но он выберет среди жесткого ограничения (жесткое соответствие) того, что он считает наиболее производительным вариантом.

Там, где все ядра обозначены как доступные, например в маске 212(P) привязки, двоичное значение может игнорироваться, потому что указывается, что программный модуль позволяет потокам выполняться любым ядром. Маска 212(P) привязки может включать в себя запланированные ядра на любых ядрах, указанных инвертированной битовой маской 208, например ядра 0, 1, 2 и 4, как представлено набором 218(P) доступных процессоров с помощью значений ядер, показывающих "1/0" (либо "1", или "0", тогда как по меньшей мере одно ядро должно иметь значение ядра "1", чтобы разрешить планирование работы в маске 212(P) привязки). В некоторых вариантах осуществления набор 218(P) доступных процессоров может выбирать ядра, которые являются работающими и простаивают при определении распределения работы доступным ядрам. В идеале, перенос работы может распределить работу ядрам 0, 1, 2 и 3, соответственно оставляя ядро 4 неиспользуемым и, по возможности, приостановленным в последующем действии. Другие соображения, которые обсуждаются ниже, могут определять, каким ядрам распределяется работа в ответ на маску 212(P) привязки, чтобы создать предпочтительное расположение. Например, предпочтительное расположение может основываться на таких факторах, как необязательная эффективность обращения к памяти.

Операции 210, 214 и 220 могут использоваться для планирования потоков, как

описано выше со ссылкой на фиг.2. К тому же в процессе 200 может выполняться другая работа с помощью операций 210, 214 и 220, например отложенный вызов процедуры (DPC), таймеры, прерывания обработки или другая работа процессора.

В соответствии с одним или несколькими вариантами осуществления, фиг.2 иллюстрирует примерное использование 222 ядер в течение временного интервала. Например, сочетание ядер, которые активны в наборе 218 доступных процессоров, может привести к использованию 222 ядер в течение заданного интервала контроля времени, например 100 миллисекунд без ограничений. Ядро 4 в использовании 222 ядер может указываться или не указываться как используемое в зависимости от того, запланирована ли работа на ядро 4 в наборе 218(P) доступных процессоров, как обсуждалось выше. С точки зрения диспетчера 114 питания ядра, использование 222 ядер в идеале будет включать в себя значение ядра "0" для ядра 4, соответственно минимизируя количество работающих ядер и приводя к сокращению энергопотребления. Независимо от использования 222 ядер, для следующего временного интервала может создаваться новая маска приостановки ядер, которая может использовать информацию из использования 222 ядер, чтобы определить новую маску приостановки ядер. В некоторых случаях ядро 3 может не потребоваться, потому что ядро 2 может иметь достаточную загрузку обработки, чтобы соответствовать второй маске 212(2) привязки, и по-прежнему является работающим ядром.

Фиг.3А и 3В показывают пояснительные использования ядер в соответствии по меньшей мере с одним вариантом осуществления раскрытия изобретения. Точнее говоря, фиг.3В показывает пояснительную модификацию использований ядер в ответ на использование ядра в пояснительной системе, которое показано на фиг.3А.

В соответствии с одним или несколькими вариантам осуществления, фиг.3А показывает систему 300, включающую некоторое количество ядер 302, 304, 306 и 308, хотя больше или меньше ядер может включаться в альтернативных вариантах осуществления системы 300. Диспетчер 114 питания ядра может управлять ядрами с использованием политики мощности. Политика мощности может определять количество активных ядер и влиять на то, как диспетчер 114 питания ядра может вычислять набор доступных ядер. Политика мощности может использоваться для ограничения количества доступных ядер или для разрешения диспетчеру 114 питания ядра масштабировать количество доступных ядер. Количество ядер, поддерживаемых платформой, может меняться у разных систем, соответственно необходима гибкая схема, чтобы позволить диспетчеру питания ядра задавать количество ядер, не зная заранее, сколько ядер доступно. Поэтому в некоторых вариантах осуществления количество ядер для использования может выражаться в виде доли от максимального использования ядра. Соответственно, реализация ядер может потребовать округления доли до следующего доступного числа, которое представляет количество ядер. Например, доля использования ядер может вычисляться как 60 процентов. Если система включает в себя четыре ядра, то доля может округляться до 75%, и три ядра могут быть работающими, в то время как одно ядро приостановлено.

Каждое ядро включает в себя использование 310 ядра ("используемость ядра"), которое представляет рабочую нагрузку этого ядра, выраженную в виде доли рабочего времени ядра в общем времени, независимо от состояния производительности ядра. Например, ядро 0 302 может иметь используемость ядра в 80%, указывающую, что ядро выполняет на 20% меньше работы, чем максимальная возможность рабочей нагрузки у ядра "0". Соответственно, используемость ядра

в 100% представляет ядро, работающее с максимальной производительностью, тогда как ядро, имеющее используемость ядра в 0%, представляет неиспользуемое ядро. В некоторых вариантах осуществления диспетчер 114 питания ядра может контролировать используемость 310 ядра.

К тому же каждое ядро может включать в себя состояние 312 производительности (р-состояние в усовершенствованном интерфейсе конфигурирования и управления энергопитанием (АСРП)). р-состояние 312 является настройкой частоты и напряжения ядра и управляется диспетчером 114 питания ядра. р-состояние 312 аналогично дроссельному регулированию в двигателе. р-состояние 312 в 100% представляет состояние максимальной производительности ядра, тогда как р-состояние в 50% представляет ядро на половине максимальной частоты с соответствующим пониженным уровнем напряжения ядра. Следует отметить, что фактическое энергопотребление ядра может не совпадать или быть пропорциональным р-состоянию 312. Например, удвоение р-состояния 312 у ядра может не удвоить энергопотребление ядра из-за других факторов, например утечки мощности ядра, базового энергопотребления ядра и/или других факторов. В некоторых вариантах осуществления диспетчер 114 питания ядра может определять и/или управлять р-состоянием 312, например путем обращения к политике мощности.

Каждое ядро включает в себя выходное значение 314 использования ("выходная используемость"), которое представляет рабочую нагрузку ядра относительно общей возможности рабочей нагрузки. Например, выходная используемость 314 может иметь размах 0-10000, где 0 представляет отсутствие использования, а 10000 представляет максимальное использование. Выходная используемость 314 может вычисляться путем умножения используемости 310 ядра и р-состояния 312. Например, ядро 0 302 включает в себя используемость ядра в 80% и р-состояние в 80%, поэтому выходная используемость равна 6400. В некоторых вариантах осуществления выходная используемость 314 используется диспетчером 114 питания ядра для принятия решений по приостановке ядра и/или определения настроек р-состояния 312, например со ссылкой на политику мощности.

В некоторых вариантах осуществления система 300 может включать в себя блок 316 ядер, например первый блок 316(1) ядер и второй блок 316(2) ядер, однако в системе 300 может быть реализовано больше или меньше блоков ядер. Блок 316 ядер может представлять собой платформу, имеющую несколько ядер с единой схемой, например двухъядерный или многоядерный процессор. Каждый блок 316 ядер может включать в себя уникальные характеристики энергопотребления. Например, ядро может включать в себя энергопотребление активного уровня, утечку ядра или другие рассеивания мощности, которые возникают, когда одно из ядер в блоке является работающим. Например, если оба ядра в первом блоке 316(1) ядер являются работающими и обладают выходной используемостью в 10000, то объединенное энергопотребление может быть 2х ватт. Если ядро 0 302 в первом блоке 316(1) ядер впоследствии приостанавливается (например, выходное использование равно 0), а ядро 1 304 остается без изменений, то объединенное энергопотребление может быть больше, чем х ватт из-за факторов, ассоциированных с блоком 316 ядер, например утечки мощности, активного энергопотребления и/или других факторов. Когда ядро 1 впоследствии приостанавливается, результирующее энергопотребление может быть равно 0 ватт. Поэтому может быть выгодным приостановить ядра так, чтобы полные блоки ядер стали приостановленными до того, как другие ядра впоследствии будут приостановлены, соответственно максимизируя энергосбережение.

Как показано на фиг.3А, пояснительная реализация системы 300 включает в себя приостанавливаемое ядро 3 308, в то время как другие ядра являются работающими (то есть активными). Хотя ядро 3 включает в себя р-состояние в 100%, это может не указывать, что энергия подается к ядру 3. Иначе говоря, диспетчер 114 питания ядра
 5 может приостановить ядро, оставляя при этом р-состояние на уровне, превышающем 0%.

В примерном сценарии использования диспетчер 114 питания ядра может вычислить выходную мощность системы 300, чтобы определить общую
 10 используемость системы в 12600 (то есть $6400+3200+3000=12600$) из максимальной общей используемости системы в 30000 (то есть $3 \text{ работающих ядер} \times 10000=30000$). Числа использования, приведенные выше, предназначены быть поясняющими по сути вычислений, которые могут выполняться с использованием общей используемости системы, и соответственно не ограничивают раскрытие изобретения.

Фиг.3В показывает пояснительную модификацию использования ядер в измененной системе 318 в ответ на общее использование системы, которое показано на фиг.3А. Измененная система 318 включает в себя изменение в приостановленном/работающем состоянии у ядер 302, 304, 306 и 308. Как отмечалось выше, со ссылкой на систему 300,
 20 общая используемость системы вычислялась равной 12600 в примере. Поэтому диспетчер 114 питания ядра может приостановить дополнительное ядро без уменьшения возможности системы соответствовать текущим потребностям рабочей нагрузки, потому что общая используемость системы меньше максимальной возможности использования у двух ядер (то есть $12600 < 20000$). В соответствии с
 25 некоторыми вариантами осуществления диспетчер 114 питания ядра может выбрать ядро для приостановки, возможно на основе политики мощности или других факторов, что может быть полностью прозрачно для пользователя системы. Прозрачность для пользователя включает в себя отсутствие воспринимаемого пользователем влияния на производительность системы или приложения, за исключением принципиальной цели, состоящей в лучшем КПД. Любые изменения, необходимые для реализации приостановки ядер, могут ограничиваться внутренними компонентами операционной системы очень низкого уровня, и соответственно не может быть поведенческого или пробного изменения для конечных пользователей,
 30 ассоциированного с приостановкой ядер.

Когда приостанавливается второе ядро, максимальная общая используемость системы у измененной системы упадет до 20000 ($2 \text{ ядра} \times 10000$). Диспетчер 114 питания ядра может приостановить любое из ядер, которые были активны в предыдущем
 40 состоянии (как показано на фиг.3А). Как обсуждалось выше, может быть выгодным приостановить ядро 2 306, чтобы полностью приостановить второй блок 316(2) ядер, что приводит к приостановленному блоку 320 ядер. Соответственно, приостановленный блок 320 ядер может увеличить энергосбережение по сравнению с приостановкой ядра 0 302 или ядра 1 304 вместо ядра 2 306.

Чтобы обеспечить прозрачное изменение для любых пользователей системы 318, выходная используемость приостановленного ядра (ядро 2) должна поглощаться или перераспределяться оставшимся работающим ядрам (то есть ядру 0 и ядру 1). Из
 45 фиг.3А выходная используемость ядра 2 была равна 3000. Поэтому в одном случае работающие ядра могут в равной степени разделить нагрузку с помощью того, что каждое ядро имеет измененную выходную используемость 322, которая предполагает дополнительную выходную используемость по 1500 для каждого ядра. В некоторых случаях могут использоваться другие разделения общей используемости
 50

приостановленного ядра при распределении используемости работающим ядрам. Например, работающее ядро может работать близко к возможности (общая используемость около 10000, что является максимальной используемостью). В таком случае ядра с большей пропускной способностью могут поглотить больше общей

используемости приостановленного ядра.

В одном или нескольких вариантах осуществления диспетчер 114 питания ядра может регулировать используемость 310 ядра у ядра 0 302 и ядра 1 304 до 79% и 47% соответственно. К тому же диспетчер 114 питания ядра может увеличить р-состояние 312 до 100% для обоих работающих ядер. Поэтому общая используемость системы у измененной системы 318 остается равной общей используемости системы у системы 300 на 12600.

Измененная система 318 иллюстрирует одно возможное изменение приостановленного/работающего состояния, используемости 310 ядра и/или р-состояния 312, чтобы обеспечить изменение ядра, которое прозрачно для пользователей, приводя при этом к сниженному энергопотреблению для измененной системы 318. Однако в системе 318 могут быть сделаны многие другие изменения, которые приводят к сниженному энергопотреблению и которые прозрачны для пользователей. Например, используемость 310 ядра у ядра 1 304 может быть увеличена до 94%, тогда как р-состояние уменьшено до 50%, приводя к общей используемости в 4700. Стратегия изменения может определяться политикой мощности, которая может учитывать конкурирующие интересы диспетчера 114 питания ядра и планировщика 116 ядра.

В политике мощности могут быть реализованы другие соображения, которые могут оказывать влияние на измененную систему 318. Например, потоки, которые не чувствительны ко времени (например, фоновые потоки), могут обладать меньшим влиянием на решение по приостановке ядер, при этом сохраняя изменения прозрачными для пользователей. Точнее говоря, с помощью включения распределения среднего времени выполнения по приоритетам потоков, выполненных на данном ядре, диспетчер 114 питания ядра может масштабировать вычисленное использование ядра таким образом, что низкоприоритетные потоки и рабочие нагрузки не учитываются так же, как высокоприоритетные потоки и рабочие нагрузки.

Фиг.4 показывает блок-схему алгоритма по меньшей мере одного варианта осуществления пояснительного процесса 400 оценивания планирования потоков и динамического использования процессоров и определения измененной конфигурации для процессоров. Процесс 400 может быть реализован диспетчером 114 питания ядра в дополнение к другим модулям в операционной системе 108 и/или находящимся в системной памяти 106.

В соответствии с одним или несколькими вариантами осуществления, диспетчер 114 питания ядра оценивает период времени для контроля выходной используемости 314 (среди прочих факторов, включающих используемость 310 ядра, р-состояние 312 и т.д.). Может выбираться период времени, который равен или длиннее временного интервала, доступного для обработки каждого потока. На заданной частоте диспетчер 114 питания ядра может начать контроль ядер.

На этапе 404 диспетчер 114 питания ядра может вычислить общее использование системы и максимальное общее использование системы. Диспетчер 114 питания ядра может оценить политику мощности на этапе 406 для определения, как регулировать использование ядер, чтобы уравновесить потребности энергосбережения у диспетчера 114 питания ядра и потребности доступности ядер (производительности

обработки) у планировщика 116 ядра. На этапе 408 другие факторы могут использоваться для определения, регулировать ли использование ядер, и если да, то как регулировать использование для достижения целей системы, например прозрачного изменения для пользователей, удовлетворения тепловых требований и/или удовлетворения других ограничений.

На этапе 410 диспетчер 114 питания ядра может вычислить новые конфигурации для ядер. Например, диспетчер питания ядра может вычислить новую битовую маску 202, которая показана на фиг.2. В одном или нескольких вариантах осуществления на этапе 410 битовая маска может регулироваться для создания набора 218 доступных процессоров. На этапе 412 диспетчер 114 питания ядра может реализовать конфигурации с этапа 410. В некоторых вариантах осуществления процесс 400 может повторяться для создания динамического распределения ядер, например путем повторения с заранее установленной частотой. В качестве альтернативы процесс 400 может создать статическое распределение ядер в течение заданного периода времени.

Иллюстративное функционирование

Фиг.5 показывает блок-схему алгоритма по меньшей мере одного варианта осуществления пояснительного процесса 500 обеспечения энергосберегающего планирования потоков и динамического использования процессоров. Процесс 500 следует толковать аналогично процессу 200 из фиг.2 в отношении упорядочения и реализации процесса. Например, порядок, в котором описываются операции в процессе 500, не предназначен для толкования в качестве ограничения, и любое количество описанных этапов может объединяться в любом порядке и/или параллельно для реализации процесса. Для целей обсуждения процесс 500 описывается со ссылкой на систему 100 из фиг.1.

В соответствии с одним или несколькими вариантами осуществления, периодическая оценочная процедура "TimeCheck" может начинаться на этапе 502. Например, отложенный вызов процедуры (DPC) может начинаться на этапе 502. В некоторых вариантах осуществления вводится конечный автомат на каждом ядре посредством DPC, работающий на фиксированной периодической частоте, сконфигурированной с помощью параметра "TimeCheck" в политике мощности для значения времени, например 100 мс, 50 мс или другого значения времени. На этапе 504 диспетчер 114 питания ядра может собрать показатели для ядер. Например, DPC ставится в очередь на каждое активное в настоящее время ядро, чтобы снять показатели для активных ядер. Показатели могут включать в себя использование ядра, распределение приоритета потоков, среднее время ожидания для готовых потоков для каждого ядра и/или показатели успеха и сбоя для пребывания в состоянии простоя, среди других возможных показателей.

На этапе 506 диспетчер 114 питания ядра может вычислить битовую маску, например маску 204 приостановки ядер. Например, новое значение может вычисляться для целевого количества активных ядер на основе пороговых величин использования, политики мощности и/или любых отношений зависимости. На этапе 508 активный набор может обновляться, например путем реализации частей процесса 200, определяемого диспетчером 114 питания ядра, чтобы создать набор 218 доступных процессоров.

На этапе 510 активный набор может быть реализован операционной системой 108. Диспетчер 114 питания ядра на этапе 512 может определить, добавлены ли ядра (работающие). Если ядра являются работающими, то на этапе 514 планировщик 116

ядра может быть уведомлен и может начать использование работающих ядер для планирования потоков. Эти работающие ядра могут быть либо целью удаленного планирования потоков (то есть от другого процессора), либо могут заранее решить выбирать потоки от других процессоров. В некоторых вариантах осуществления

5 потоки могут перераспределяться для выполнения на работающих ядрах, посредством этого уменьшая рабочую нагрузку с других работающих ядер.

На этапе 516 предполагаемое выходное использование ядра может вычисляться диспетчером 114 питания ядра и может включать в себя работающие ядра с этапа 506.

10 Таким образом, новое значение для количества активных ядер, определенное на этапе 506, используется для вычисления предполагаемого использования процессора. На этапе 518 диспетчер 114 питания ядра может вычислить новое значение для р-состояния. В некоторых вариантах осуществления DPC назначается на каждое ядро, чтобы обновить целевое р-состояние. В примере, если предполагаемое использование

15 увеличивается, то значение р-состояния также может увеличиваться, если никакие ядра не работают на этапе 512. Однако если ядра являются работающими на этапе 512, то р-состояние может увеличиваться или уменьшаться для уравнивания потребностей энергосбережения у диспетчера 114 питания ядра с потребностями обработки у

20 планировщика 116 ядра.

На этапе 520 диспетчер 114 питания ядра определяет, следует ли изменять текущие значения р-состояния и/или t-состояния (состояние линейной регулировки), на основе результатов вычисления на этапе 518. Если на этапе 520 изменяются р-состояние и/или t-состояние, то диспетчер 114 питания ядра может поставить в очередь

25 переходные DPC на соответствующие ядра на этапе 522. Таким образом, запланированные планировщиком 116 ядра потоки могут назначаться ядрам, работающим в новом р-состоянии и/или t-состоянии, как реализовано на этапе 520.

На этапе 524 диспетчер 114 питания ядра определяет, приостановлены ли ядра на

30 этапе 506. Например, предполагаемое выходное использование ядра с этапа 516 может быть меньше текущего использования. Если ядра приостановлены на этапе 524, то диспетчер 114 питания ядра может уведомить планировщика 116 ядра на этапе 526, чтобы прекратить планирование потоков на недавно приостановленные ядра. Для любых ядер, добавленных или удаленных из маски активных ядер, планируется DPC

35 для этого ядра. Как описано со ссылкой на фиг.3А и 3В, работа с приостановленного ядра может перераспределяться одному или нескольким работающим ядрам. В некоторых вариантах осуществления недавно приостановленные ядра могут быть введены в самое глубокое С-состояние, которое доступно. В конечном счете на

40 этапе 528 процесс 500 может повторяться.

Фиг.6 показывает другую блок-схему алгоритма по меньшей мере одного варианта осуществления пояснительного процесса 600 обеспечения энергосберегающего планирования потоков и динамического использования процессоров, дополнительно

45 включающего учет простоя доменов. Процесс 600 включает в себя многие из подпроцессов, которые описаны на фиг.5, и поэтому те подпроцессы из фиг.5 не будут описываться снова.

На этапе 602 диспетчер 114 питания ядра может определить, задействован ли учет простоя доменов. Если учет простоя доменов задействован, то на этапе 604 владелец домена снимает показатели для домена (например, блока 316 ядер или всех ядер).

50

В некоторых вариантах осуществления учет простоя доменов может инициировать другое решение на этапе 606. Если задействован учет простоя доменов, то диспетчер 114 питания ядра может вычислить целевое р-состояние домена на

этапе 608. В конечном счете на этапе 610 процесс 600 может повторяться.

Фиг.7 показывает блок-схему алгоритма по меньшей мере одного варианта осуществления процесса 700 предоставления дополнительных настроек политики мощности и других входных данных для энергосберегающего планирования потоков и динамического использования процессоров. В некоторых вариантах осуществления приостановка ядер может быть реализована в виде расширения к существующему конечному автомату, который вычисляет целевое состояние, которое нужно использовать для состояний производительности процессора. Точное количество ядер, которое должно использоваться в любое заданное время, будет определяться на основе некоторого количества факторов, которые описаны ниже со ссылкой на процесс 700.

В соответствии с одним или несколькими вариантами осуществления, текущая политика мощности может использоваться для задания количества ядер для использования на этапе 702. На этапе 704 может задаваться минимальное или максимальное количество ядер. Диспетчер 114 питания ядра может вычислить количество активных ядер, необходимое для выполнения заданной рабочей нагрузки энергосберегающим способом. В некоторых случаях запуск минимального количества ядер может быть выгодным для энергосбережения и помогает диспетчеру 114 питания ядра. Наоборот, запуск максимального количества ядер может обеспечивать наивысший уровень производительности, соответственно помогая планировщику 116 ядра.

На этапе 706 дополнительные параметры политики мощности для приостановки ядер могут быть реализованы диспетчером 114 питания ядра с использованием одного или нескольких подпроцессов 706(1), ..., 706(4). На этапе 706(1) может регулироваться необходимый интервал времени для приостановки и/или возобновления работы ядер. Например, частота приостановки и/или возобновления работы ядер может управляться путем изменения интервала. В некоторых вариантах осуществления ядра могут приостанавливаться в первом интервале и возобновлять работу во втором интервале. Например, политика, способствующая энергосбережению, может приостанавливать ядра каждые 100 мс, но может возобновлять работу ядер каждые 500 мс.

На этапе 706(2) диспетчер 114 питания ядра может реализовать политику увеличения и/или уменьшения. Например, первый вариант политики может единовременно приостанавливать только установленное количество ядер, например одно ядро единовременно. Второй вариант политики может приостанавливать или возобновлять работу ядер для достижения идеального использования ядра, соответственно единовременно приостанавливая и/или возобновляя работу нескольких ядер. Третья политика может ударяться из одной крайности в другую (либо приостанавливать как можно больше, либо возобновлять как можно больше).

На этапе 706 (3) необходимая пороговая величина использования может быть увеличена или уменьшена на основе занятости процессоров. Например, процессор может не подвергаться изменению в приостановленном или работающем состоянии, пока процессор (или другие процессоры) не включит состояние занятости для заданного периода времени. Это может уменьшить переключение процессора между приостановленным и работающим состоянием в быстрой последовательности.

В конечном счете, на этапе 706(4) может быть реализована политика для масштабирования работающих ядер. Например, диспетчер 114 питания ядра может вычислить идеальное состояние производительности целевого процессора на основе

количества ядер процессора в активном наборе в настоящее время. Чтобы обеспечить наилучший компромисс между энергосбережением, производительностью и реакцией на определенные рабочие нагрузки, диспетчер 114 питания ядра может преимущественно запустить меньшее количество процессоров в состоянии с большей
 5 производительностью, или наоборот, запустить большее количество ядер в состоянии с меньшей производительностью.

На этапе 708 эвристические правила ядра и/или системы могут быть реализованы диспетчером 114 питания ядра с использованием одного или нескольких
 10 подпроцессов 708(1) и 708(2). На этапе 708(1) диспетчер 114 питания ядра может вычислить необходимое количество активных ядер и оптимальное состояние производительности у ядер при активном использовании на основе успешного использования глубоких состояний простоя процессора (состояний ожидания). Это может позволить диспетчеру питания ядра обнаружить, когда более глубокие
 15 состояния простоя не используются эффективно в наборе активных (работающих) ядер. Чтобы сберегать энергию и по-прежнему обеспечивать производительность, может быть выгодным помещать больше ядер в приостановленное состояние и увеличивать состояние производительности у оставшихся активных ядер, чтобы
 20 гарантировать, что работа выполняется эффективно.

На этапе 708(2) среднее время ожидания может использоваться для потоков, готовых для распределения (назначения) ядру. Например, диспетчер 114 питания ядра может вычислить необходимое количество ядер процессора, используя распределение
 25 среднего времени ожидания для потоков в состоянии готовности, что позволяет диспетчеру питания ядра масштабировать количество используемых ядер, чтобы уменьшить задержку перед тем, как потоки в состоянии готовности могут выполняться, соответственно увеличивая производительность и реакцию. В примере, когда нужно выполнить большое количество потоков, может быть выгодным
 30 восстановить работу ядер наряду с уменьшением р-состояния ядер, так как каждое ядро одновременно может обрабатывать только один поток. Таким образом, в некоторых случаях большая доступность ядер даст возможность обработки большего количества потоков.

На этапе 710 зависимости состояния простоя могут использоваться диспетчером 114
 35 питания ядра для настройки реализации приостановки ядер. Когда диспетчер 114 питания ядра выбирает, какие определенные ядра следует приостановить или вернуть к работе, он сначала изучит отношения зависимости состояния простоя у ядер, чтобы определить, какие ядра могли бы совместно использовать ресурсы энергии или
 40 тактирования, и выберет приостановку или возобновление ядер наиболее энергоэффективным способом на основе совместно используемых средств управления.

На этапе 712 отношения состояния производительности и регулировки могут учитываться при определении того, следует ли приостановить или возобновить работу
 45 одного или нескольких ядер. Ядра, которые совместно используют состояние производительности или элементы регулирования, могут приостанавливаться или возобновляться вместе, чтобы реализовать более высокие КПД.

На этапе 714 могут рассматриваться отношения комплектов (блоков) ядер, например зависимости, описанные на фиг.3А и 3В касательно используемостей блока
 50 ядер, а точнее говоря, приостановка блока ядер перед приостановкой другого ядра в новом блоке ядер. В конечном счете, на этапе 716 локальность памяти может использоваться диспетчером 114 питания ядра при реализации соображений приостановки ядер. Например, два или более ядра могут иметь связи по комплекту,

например ядра, совместно использующие физический корпус процессора, имеющий группу блоков совместно используемой памяти (например, узел NUMA (доступ к неоднородной памяти)). Группа блоков совместно используемой памяти может позволить ядрам иметь сокращенное время обращения к памяти по сравнению с ядрами, которые совместно не используют группу блоков совместно используемой памяти.

Заключение

Вышеописанные методики, системы и устройства имеют отношение к обеспечению энергосберегающего планирования потоков и динамического использования процессоров. Несмотря на то что методики, системы и устройства описаны на языке, характерном для структурных признаков и/или методологических действий, необходимо понимать, что прилагаемая формула не обязательно ограничивается описанными характерными признаками или действиями. Точнее, характерные признаки и действия раскрываются в качестве типовых форм реализации таких методик и устройств.

Формула изобретения

1. Способ уравнивания производительности и энергосбережения вычислительного устройства, имеющего множество ядер, содержащий этапы, на которых:

определяют, какие ядра из упомянутого множества ядер активно выполняют работу;

определяют политику мощности, чтобы запустить план производительности и энергосбережения для упомянутого множества ядер, при этом при определении политики мощности:

создают маску приостановки ядер с использованием битового значения для представления приостановленного или работающего состояния ядра,

определяют маски привязки потока к процессору, представляющие одно или более ядер, которым назначена обработка потока;

обеспечивают, по меньшей мере, часть плана производительности и энергосбережения для ядер путем объединения инвертированной маски приостановки ядер и масок привязки потока к процессору, используя оператор "И", чтобы создать набор доступных процессоров, и

вычисляют, какие ядра обозначаются как приостановленные или работающие, на основе, по меньшей мере частично, набора доступных процессоров; и

приостанавливают по меньшей мере одно из ядер, активно выполняющих работу, на основе, по меньшей мере частично, политики мощности, указывающей, что это по меньшей мере одно из ядер, активно выполняющих работу, обозначено как приостановленное ядро.

2. Способ по п.1, в котором определение политики мощности совершается динамически в виде итеративного процесса.

3. Способ по п.1, дополнительно содержащий этап, на котором масштабируют по меньшей мере одно из упомянутого множества ядер на основе политики мощности.

4. Способ по п.3, в котором при масштабировании по меньшей мере одного из упомянутого множества ядер регулируют по меньшей мере одно из используемости ядра и состояния питания (р-состояния) ядра, чтобы повысить энергосбережение.

5. Способ по п.1, в котором при приостановке по меньшей мере одного из упомянутого множества ядер:

определяют, включает ли в себя работающий блок ядер приостановленное ядро; и если определено, что работающий блок ядер имеет приостановленное ядро, приостанавливают по меньшей мере одно работающее ядро в работающем блоке ядер.

6. Способ по п.1, дополнительно содержащий этап, на котором изменяют политику мощности с использованием эвристических правил ядра и системы.

7. Способ по п.1, дополнительно содержащий этап, на котором изменяют политику мощности с использованием взаимоотношений между ядрами в виде зависимостей состояния простоя процессора.

8. Машиночитаемый носитель информации, содержащий машиноисполняемые команды, которыми при их исполнении процессором выполняются действия, в соответствии с которыми:

контролируют активность ядер в многоядерной системе;

получают политику мощности для многоядерной системы, причем политика мощности уравнивает энергосбережение и производительность обработки для каждого ядра в многоядерной системе;

приостанавливают по меньшей мере одно ядро в ответ на активность ядер на основе политики мощности;

изменяют контролируемую активность ядер для создания измененной активности ядер;

возобновляют работу по меньшей мере одного ядра в ответ на измененную активность ядер на основе политики мощности; и

переназначают по меньшей мере один поток этому по меньшей мере одному работающему ядру.

9. Машиночитаемый носитель информации по п.8, в котором при приостановке ядра снижают состояние питания ядра до состояния простоя процессора с минимальной мощностью (C-состояние).

10. Машиночитаемый носитель информации по п.8, дополнительно содержащий регулировку состояния питания по меньшей мере одного работающего ядра в многоядерной системе.

11. Машиночитаемый носитель информации по п.8, в котором приостановка по меньшей мере одного ядра в ответ на активность ядер запускается динамически с заранее установленной частотой.

12. Машиночитаемый носитель информации по п.8, в котором контроль активности ядер в многоядерной системе совершается в течение заранее установленного периода времени, при этом приостановка по меньшей мере одного ядра в ответ на активность ядер совершается динамически в ответ на возможность энергосбережения, обнаруженную посредством контролируемой активности ядер.

13. Машиночитаемый носитель информации по п.8, в котором при приостановке по меньшей мере одного ядра в ответ на активность ядер переносят работу на работающее ядро, что разрешено маской привязки, ассоциированной с планом работы программ.

14. Машиночитаемый носитель информации по п.8, в котором при возобновлении работы по меньшей мере одного ядра возобновляют работу ядра в блоке ядер, который включает в себя по меньшей мере одно другое работающее ядро, причем этот блок ядер имеет некоторый уровень рассеивания мощности, когда какие-либо из ядер в данном блоке ядер являются работающими.

15. Система с множеством логических процессоров, выполненная с возможностью уравнивания ее производительности и энергосбережения, содержащая:

множество процессоров, причем это множество процессоров включает в себя по меньшей мере один блок ядер, имеющий несколько ядер; и

контроллер, соединенный с множеством процессоров, при этом контроллер предназначен для:

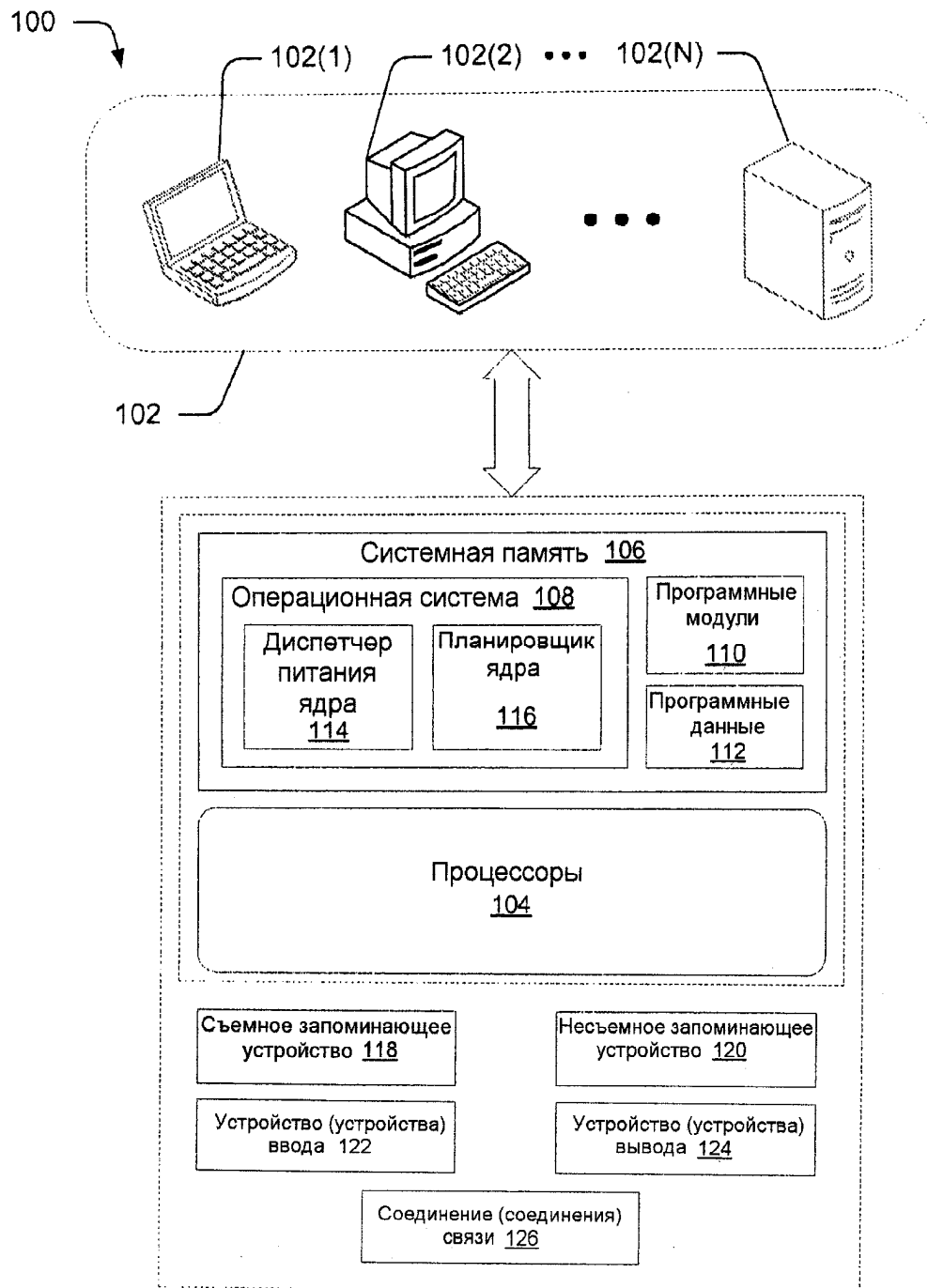
5 реализации плана производительности,
реализации политики энергосбережения, которая включает в себя маску
приостановки ядер, причем маска приостановки ядер объединяется с планами работы
программных модулей для определения результирующего набора процессоров, на
10 который нужно планировать работу; и

уравновешивания плана производительности и политики энергосбережения путем
приостановки одного или более из упомянутого множества процессоров и создания
приоритизации приостановки ядер для работающего ядра в блоке ядер, имеющем по
15 меньшей мере одно приостановленное ядро.

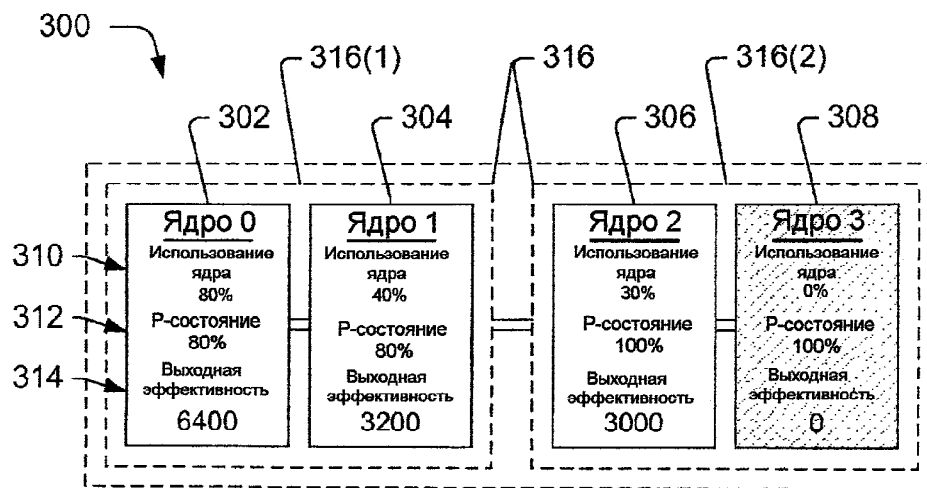
16. Система по п.15, в которой контроллер принимает команды от диспетчера
питания ядра, находящегося в системной памяти, чтобы реализовать политику
энергосбережения.

17. Система по п.15, дополнительно реализующая регулирование состояния
20 питания (p-состояния) по меньшей мере одного работающего ядра.

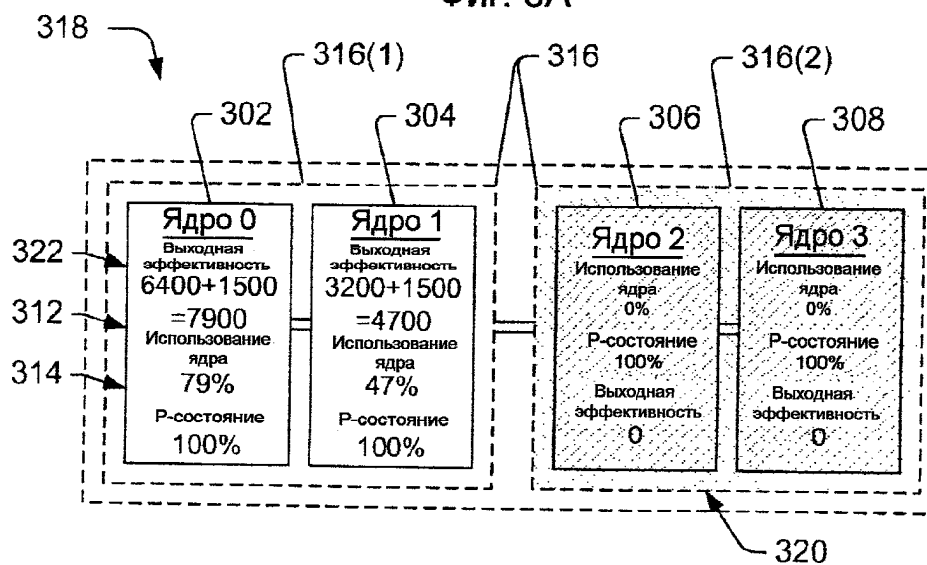
18. Система по п.15, в которой политика энергосбережения ограничивает
количество используемых ядер, что выражается как процентная доля от
максимального использования ядер.



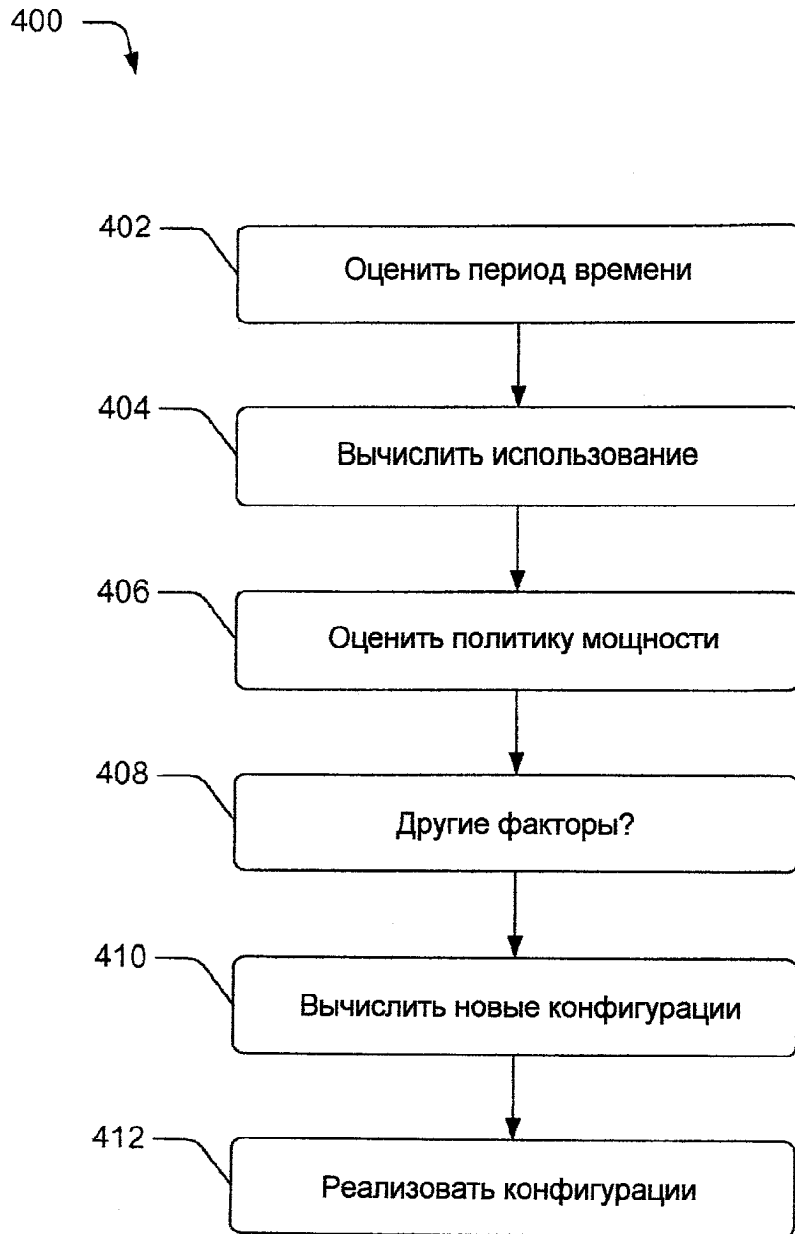
Фиг. 1



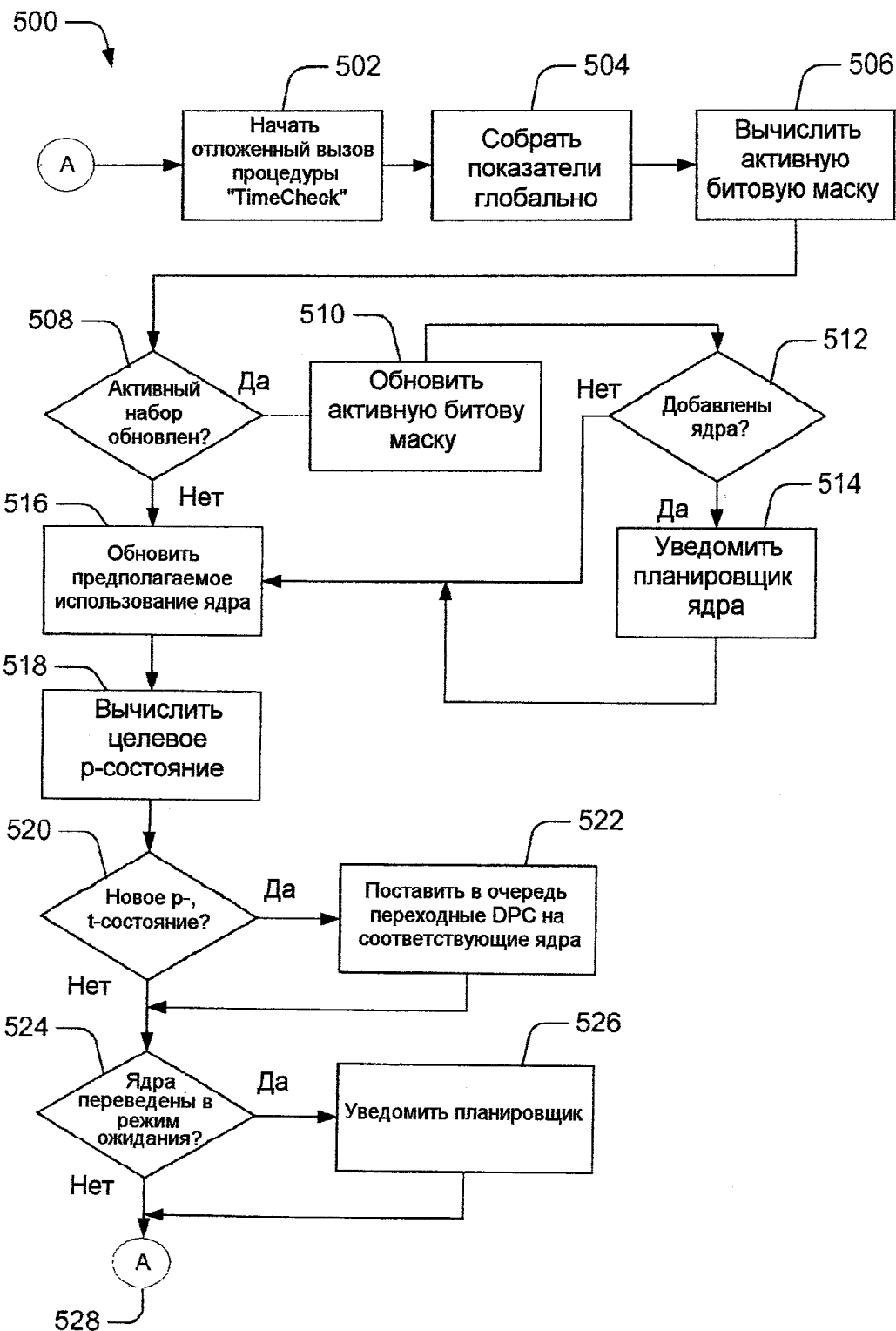
Фиг. 3А



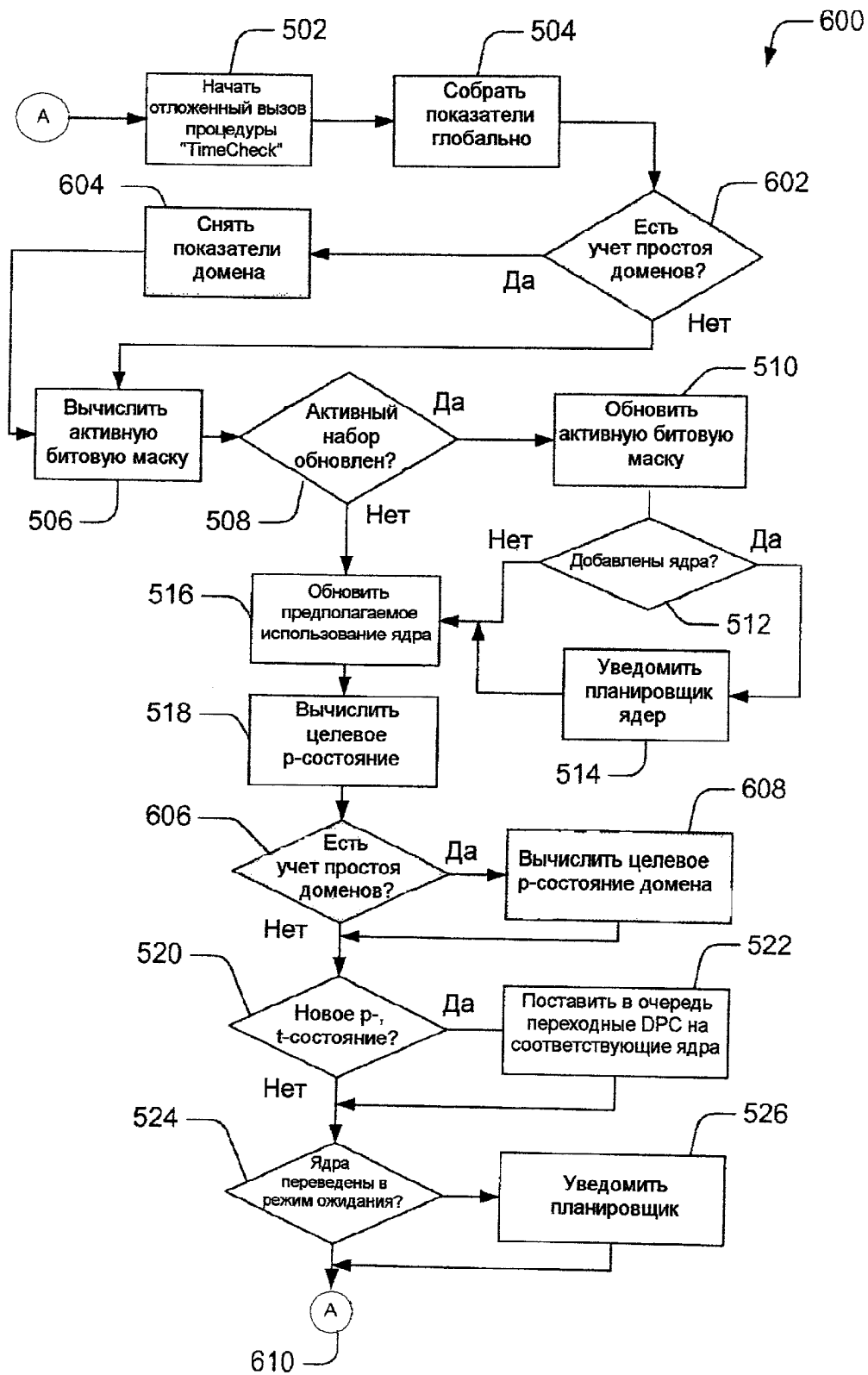
Фиг. 3В



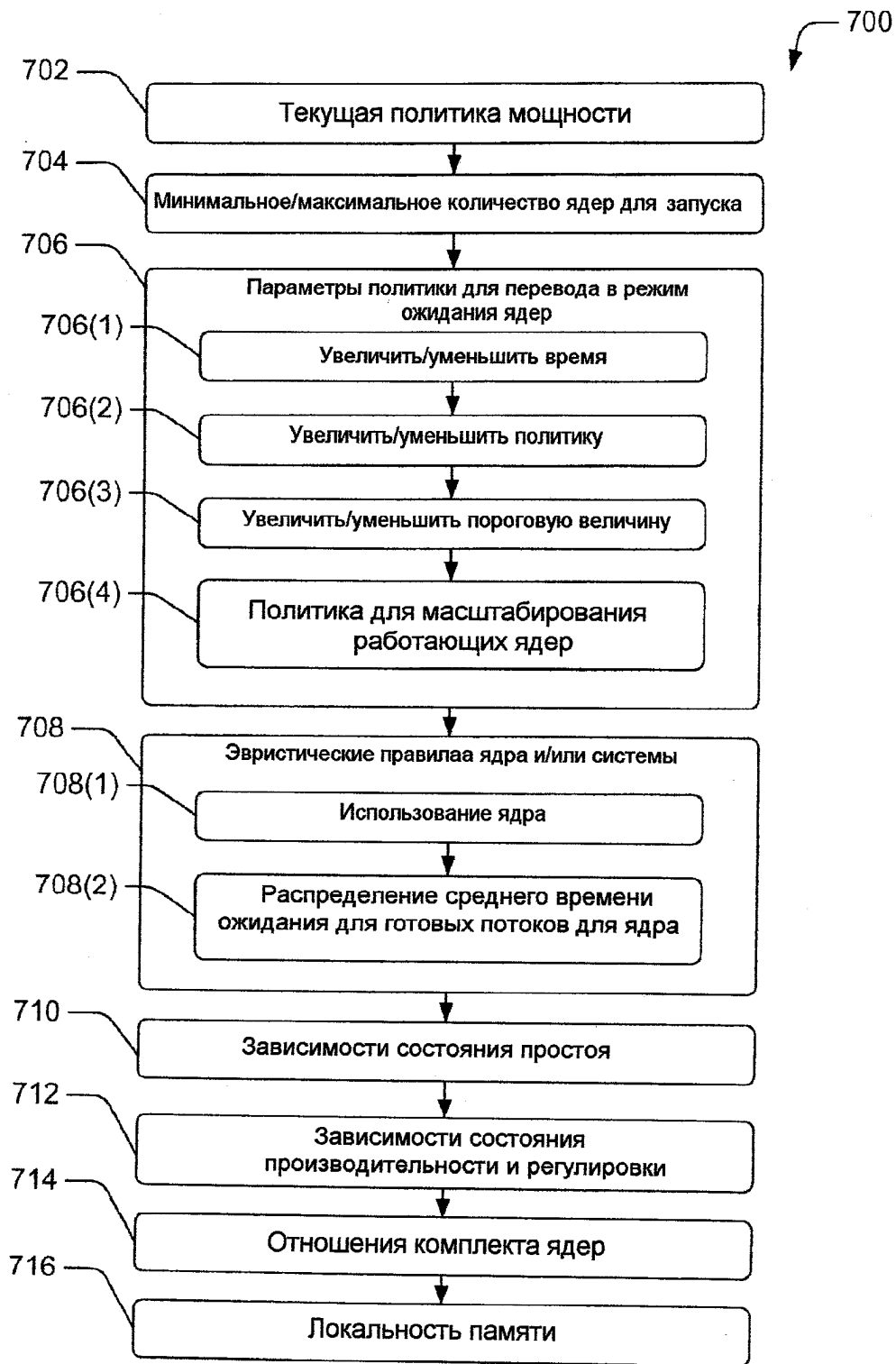
Фиг. 4



Фиг. 5



Фиг. 6



Фиг. 7