



**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ(21)(22) Заявка: **2010148134/08**, **13.04.2009**(24) Дата начала отсчета срока действия патента:
13.04.2009

Приоритет(ы):

(30) Конвенционный приоритет:
26.05.2008 US 12/127,006(43) Дата публикации заявки: **27.05.2012** Бюл. № 15(45) Опубликовано: **20.02.2014** Бюл. № 5(56) Список документов, цитированных в отчете о
поиске: **US 2006/0015490 A1, 19.01.2006. US**
2005/0080757 A1, 14.04.2005. US 2006/0085440
A1, 20.04.2006. US 2005/0182775 A1, 18.08.2005.
RU 2313824 C2, 27.12.2007.(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: **25.11.2010**(86) Заявка РСТ:
US 2009/040296 (13.04.2009)(87) Публикация заявки РСТ:
WO 2009/151763 (17.12.2009)

Адрес для переписки:

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

(72) Автор(ы):

**КУНЕО Эндрю Р. (US),
УОРЛАЙН Бен (US),
ЦЕНЦ Эрик М. (US),
ЧЖУ Шон (US)**

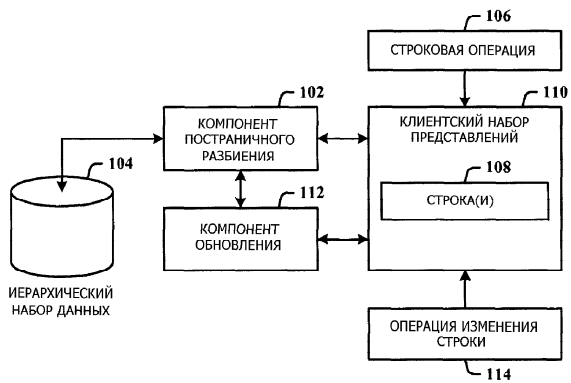
(73) Патентообладатель(и):

МАЙКРОСОФТ КОРПОРЕЙШН (US)**(54) ПОСТРАНИЧНОЕ РАЗБИЕНИЕ ИЕРАРХИЧЕСКИХ ДАННЫХ**

(57) Реферат:

Изобретение относится к области обработки иерархических наборов данных. Техническим результатом является оптимизация производительности операций с иерархическими наборами данных в клиент-серверном окружении посредством минимизации объема данных, посылаемых между клиентом и сервером, при считывании или редактировании иерархических наборов данных. Раскрыта архитектура, которая обеспечивает возможность интеллектуальным образом постранично разбивать (расслаивать)

иерархические наборы данных с помощью операций представления иерархических представлений. Она дополнительно обеспечивает возможность делать дополнения к/удаления из иерархических представлений и управлять отношениями "родитель/потомок" записей набора данных, не посылая весь набор записей клиенту и не принимая весь набор записей обратно на сервере, тем самым оптимизируя производительность операций с иерархическими наборами данных. 3 н. и 17 з.п. ф-лы, 8 ил.



Фиг. 1

RU 2507574 C2

RU 2507574 C2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.
G06F 17/30 (2006.01)

(12) **ABSTRACT OF INVENTION**

(21)(22) Application: **2010148134/08, 13.04.2009**
 (24) Effective date for property rights:
13.04.2009
 Priority:
 (30) Convention priority:
26.05.2008 US 12/127,006
 (43) Application published: **27.05.2012 Bull. 15**
 (45) Date of publication: **20.02.2014 Bull. 5**
 (85) Commencement of national phase: **25.11.2010**
 (86) PCT application:
US 2009/040296 (13.04.2009)
 (87) PCT publication:
WO 2009/151763 (17.12.2009)
 Mail address:
129090, Moskva, ul. B. Spasskaja, 25, str.3, OOO "Juridicheskaja firma Gorodisskij i Partnery"

(72) Inventor(s):
**KUNEO Ehndrju R. (US),
UORLAJN Ben (US),
TsENTs Ehrik M. (US),
ChZhU Shon (US)**
 (73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)

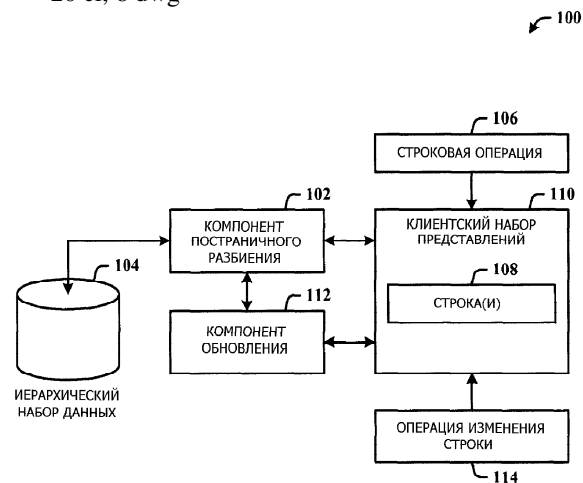
RU 2 507 574 C2

RU 2 507 574 C2

(54) **PAGE-BY-PAGE BREAKDOWN OF HIERARCHICAL DATA**

(57) Abstract:
 FIELD: information technology.
 SUBSTANCE: disclosed is an architecture which facilitates intelligent page-by-page breakdown (splitting) of hierarchical data sets using hierarchical presentation operations. The architecture further enables to add to/remove from hierarchical presentations and manage "parent/child" relationships of data set records without sending the whole set of records to a client and without receiving the whole set of records back at the server, thereby optimising efficiency of operations over hierarchical data sets.
 EFFECT: optimising efficiency of operations over hierarchical data sets in a client-server environment by minimising the amount of data sent between a client and a server when reading or editing

hierarchical data sets.
20 cl, 8 dwg



Фиг. 1

Предшествующий уровень техники

Повсеместный характер сетей и систем данных облегчает широкий доступ к большим объемам данных. Предприятия, например, теперь поддерживают распределенные, так же, как и локальные, системы данных, которые хранят все виды данных, относящихся, например, к финансовой деятельности, продажам, продуктам, управлению проектами, человеческим ресурсам и так далее. Таким образом, способность пользователя осуществить доступ к данным серверных систем может влиять на производительность серверов, сетей и клиентских систем.

Традиционные системы данных, как правило, используют запрашивание и извлечение больших наборов данных. Трудно поддерживать интерактивное представление и редактирование больших иерархических наборов данных в клиент-серверной системе таким способом, который развивает положительный пользовательский опыт. Сортировка, группировка и другие операции представления, для которых необходим весь набор данных, являются очень ресурсоемкими и плохо масштабируемыми на сервере с ростом числа пользователей. Кроме того, по мере того, как в уровни иерархии добавляются записи, эти записи также должны учитываться в вышеупомянутых операциях. При совмещении этих ограничений с добавочной сложностью иерархического набора данных (например, проект, который содержит несколько уровней задач) сложность проблемы увеличивается, и производительность на клиенте и/или сервере страдает.

Сущность изобретения

Далее представлено краткое изложение сущности изобретения для того, чтобы обеспечить базовое понимание некоторых новых вариантов осуществления, описанных в данном документе. Настоящий раздел не является широким обзором и не предназначен для определения ключевых/критических элементов или очерчивания их объема. Его исключительной целью является представление некоторых концепций в упрощенной форме в качестве предисловия к более подробному описанию, которое представлено ниже.

Раскрытая архитектура оптимизирует производительность операций с иерархическими наборами данных в клиент-серверном окружении посредством минимизации объема данных, посылаемых между клиентом и сервером, при считывании или редактировании иерархических наборов данных. Архитектура обеспечивает возможность интеллектуальным образом постранично разбивать иерархические наборы данных с помощью операций представления (например, сортировки, фильтрации, группировки), возможность делать дополнения к/удаления из иерархии и управлять отношениями "родитель/потомок" записей набора данных, не посылая весь набор записей клиенту или обратно серверу. Управление может быть оптимизировано для операций считывания, но также минимизирует данные, передаваемые во время операций редактирования.

В завершение вышесказанного, конкретные иллюстративные аспекты описываются в данном документе совместно с последующим описанием и прилагаемыми чертежами. Эти аспекты указывают различные способы, которыми раскрытые в данном документе принципы могут быть осуществлены на практике, все аспекты и эквиваленты которых также рассматриваются в объеме заявленного изобретения. Другие преимущества и новые признаки станут очевидными из последующего подробного описания при рассмотрении совместно с чертежами.

Краткое описание чертежей

Фиг.1 иллюстрирует компьютерно-реализованную систему обработки данных

согласно раскрытой архитектуре.

Фиг.2 иллюстрирует клиент-серверную систему для постраничного разбиения иерархических данных.

Фиг.3 иллюстрирует механизм отслеживания, используемый на клиенте совместно с операциями, выполняемыми с иерархическим представлением.

Фиг.4 иллюстрирует развернутое представление и свернутое представление строк и информацию отслеживания в индексе конвертера позиций и список свернутых наборов.

Фиг.5 иллюстрирует способ обработки данных.

Фиг.6 иллюстрирует способ отслеживания операций представления на стороне клиента.

Фиг.7 иллюстрирует блок-схему вычислительной системы, способной выполнять процессы постраничного разбиения и представления согласно раскрытой архитектуре.

Фиг.8 иллюстрирует блок-схему примерного вычислительного окружения для выполнения процессов постраничного разбиения и представления.

Подробное описание

Раскрытая архитектура является клиент-серверной реализацией, которая обеспечивает способностью интеллектуальным образом постранично разбить (разделить на части) иерархические наборы данных с помощью операций представления иерархического представления на клиенте. Это дополнительно обеспечивает способность делать дополнения к/удаления из иерархического представления на клиенте и управлять отношениями "родитель/потомок" записей представления, никогда не посылая весь набор записей клиенту и не принимая весь набор записей обратно на сервере, тем самым оптимизируя производительность операций с иерархическими наборами данных. Архитектура максимизирует объем данных, которые могут быть постранично разбиты, и минимизирует данные, которые не могут быть постранично разбиты.

Последующая терминология, относящаяся к представлению, используется в описании сквозным образом.

Представление набора данных является поднабором большего набора данных, определяемым поднабором полей, поднабором строк, критерием фильтрации строк, критерием группировки, критерием сортировки; представление набора данных существует только на сервере. Представление строк является в точности поднабором строк, упомянутым в представлении набора данных. Этот список строк является "полностью развернутым" списком. Индекс в этом списке является тем, что называют "позицией данных". Представление строк существует на сервере и на клиенте.

Клиентское представление является поднабором представления строк, которое не включает потомков свернутых строк. Индекс в этом списке является тем, что называют 'позицией представления'. Клиентский набор представлений является подмножеством клиентского представления и является тем, что пользователь может физически видеть на клиенте в данный момент. Таким образом, если клиентское представление содержит 100 строк, и 30 строк были загружены на клиент, но только первые 10 строк видимы на экране без прокрутки, то клиентский набор представлений имеет размер 10.

Далее ссылка будет сделана на чертежи, где подобные ссылочные позиции используются для ссылки на подобные элементы сквозным образом. В последующем описании, в целях объяснения, сформулированы несколько конкретных подробностей, чтобы обеспечить полное их понимание. Может быть очевидно, однако, что новые варианты осуществления могут быть реализованы на практике без этих конкретных

подробностей. В других случаях известные структуры и устройства показаны в форме блок-схемы, чтобы облегчить их описание. Намерение состоит в том, чтобы охватить все модификации, эквиваленты и альтернативы, находящиеся в пределах духа и объема заявленного изобретения.

5 Фиг.1 иллюстрирует компьютерно-реализованную систему 100 обработки данных согласно раскрытой архитектуре. Система 100 включает в себя компонент 102 постраничного разбиения для постраничного разбиения (получения поднабора) иерархического набора 104 данных на основании строковой операции 106 с одной или 10 более строками 108 в иерархическом клиентском наборе 110 представлений. Система 100 также включает в себя компонент 112 обновления для того, чтобы обновлять иерархический клиентский набор 110 представлений согласно операции 114 изменения строки. Обновление клиентского набора 110 представлений может быть 15 основано на строковых ключах представления строк, причем строковые ключи связаны со страницей иерархического набора 104 данных. Компонент 102 постраничного разбиения может постранично разбивать иерархический набор 104 данных асинхронно на основании операции 114 изменения строки.

Отметим, что операция 114 изменения строки является строковой операцией, 20 которая имеет характер редактирования по отношению к иерархии, существующей в представлении 206 набора данных. Операция 114 изменения строки является вставкой строки, удалением строки или формированием смещения и/или расформированием смещения строки. Операция 114 изменения строки может привести к последующей строковой операции 106, если становится необходимо обновить клиентский набор 110 25 представлений строками, которые были прокручены в представление в результате удаления строки, например.

Система 100 минимизирует объем обрабатываемых данных, обрабатывая только релевантные строки в представлении строк вместо того, чтобы повторно вычислять 30 все представление строк. С этой целью строковая операция 106 может быть операцией считывания, которая сворачивает строки-потомки иерархического клиентского набора 110 представлений в свернутую строку, разворачивает свернутую строку, чтобы показать одну или более новых строк-потомков в иерархическом клиентском наборе 110 представлений, и добавляет или удаляет строки на основании 35 прокручивания иерархического клиентского набора 110 представлений. Операции считывания могут быть выполнены во время режима считывания, когда иерархические данные посылаются клиенту только для чтения. В этом режиме не разрешено редактирование, и по этой причине грид может быть более экономичным в 40 отношении объема не разбитых постранично данных (например, нефильТРованная иерархия), которые грид посылает клиенту. Грид является вычислительной инфраструктурой для предоставления больших объемов данных локальных и/или распределенных систем данных.

Прокрутка может привести к тому, что строки будут прикрепляться к клиентскому 45 набору 110 представлений или удаляться из клиентского набора 110 представлений. Например, при прокрутке вверх строки будут добавляться сверху и удаляться снизу клиентского набора 110 представлений. Добавление строк сверху клиентского набора 110 представлений приводит к тому, что новые запросы посылаются в 50 компонент 102 постраничного разбиения для постраничного разбиения набора 104 данных для желаемых новых строк.

Строковая операция может быть операцией редактирования, связанной со вставкой строки, удалением строки или формированием смещения и/или расформированием

смещения строки в иерархическом клиентском наборе 110 представлений. Операции редактирования могут быть выполнены в режиме редактирования, когда иерархические данные посылаются клиенту в формате, который поддерживает операции редактирования строк. Формирование смещения является действием увеличения структурного уровня для строки, чтобы сделать строку дочерней для другой строки. Расформирование смещения является действием уменьшения структурного уровня для строки, чтобы сделать строку одноуровневой с ее родительской строкой.

Компонент 102 постраничного разбиения постранично разбивает иерархический набор 104 данных на строки, релевантные для строковой операции, на основании фильтрации, сортировки, разворачивания строки и сворачивания строки, например.

Фиг.2 иллюстрирует клиент-серверную систему 200 для постраничного разбиения иерархических данных. Система 200 воплощает компонент 102 постраничного разбиения и компонент 112 обновления как часть серверной системы 202 для постраничного разбиения иерархического набора 104 данных серверной системы, и иерархический клиентский набор 110 представлений, строку(и) 108 представления, строковую операцию 106 и операцию 114 изменения строки, как они имеют место на клиенте 204. Компонент 112 обновления обновляет неотфильтрованную иерархию на основании строковых операций, связанных с режимом редактирования.

Система 200 дополнительно иллюстрирует данные, сгенерированные и сохраненные как на клиенте 204, так и на сервере 202. Компонент 102 постраничного разбиения генерирует представление 206 набора данных и представление 208 строк стороны сервера. Копия представления 208 строк затем посылается клиенту 204 как представление 210 строк стороны клиента, чтобы минимизировать объем данных, которые иначе должны были бы быть посланы между сервером 202 и клиентом 204. Клиентское представление 212 может затем быть получено из представления 210 строк стороны клиента.

Компонент 102 постраничного разбиения управляет отношениями "родитель/потомок" строк в иерархическом клиентском наборе 110 представлений, которые релевантны строковой операции 106 и/или операции 114 изменения строки. Компонент 102 постраничного разбиения также создает индекс структуры иерархии (представление 208 строк стороны сервера), определяемый упорядоченным отображением строковых ключей и родительских ключей. Индекс затем посылается клиенту 204 на основании строковой операции (например, строковой операции 106, операции 114 изменения строки) иерархического клиентского набора 110 представлений. Эти и другие аспекты будут описаны более подробно в данном документе ниже.

Фиг.3 иллюстрирует механизм отслеживания, используемый на клиенте 204 совместно с операциями, выполняемыми в отношении иерархического клиентского набора 110 представлений. Здесь, различные типы строковых операций 300 (например, только для чтения, редактирования) могут быть выполнены со строкой(ами) 108 из клиентского набора 110 представлений. Чтобы отследить активность строки в основанном на клиенте клиентском наборе 110 представлений, используются индекс 302 конвертера позиции и список 304 свернутых наборов. Индекс 302 конвертера позиции является упорядоченным индексом строк, который используется для отслеживания развернутого/свернутого состояния иерархического клиентского набора 110 представлений для трансляции между позицией представления и позицией данных. Позиция представления является порядковой позицией строки в пределах

клиентского набора 110 представлений относительно развернутых/свернутых строк. Позиция данных (также называемая индексом строки) является порядковой позицией строки в пределах клиентского набора 110 представлений, который включает в себя все строки, независимо от того, развернуты строки или нет. Список 304 свернутых наборов является списком свернутых строк, который позволяет гриду знать, будет ли строка свернута или развернута. Список 304 свернутых наборов первоначально пуст, но поскольку строки разворачиваются/сворачиваются на клиенте 106, этот список 304 обновляется.

Фиг.4 иллюстрирует развернутое представление 400 и свернутое представление 402 строк и информацию отслеживания в индексе конвертера позиции и списке свернутых наборов. Архитектура минимизирует объем данных, который посылается между клиентом и сервером при считывании или редактировании иерархических наборов данных. Иллюстративный пример описывает операции считывания, однако данные, передаваемые во время операций редактирования, могут быть также минимизированы. Операции, как чтения, так и редактирования, начинаются с запроса от клиента к серверу данных, которые будут воспроизведены гридом. Сервер опрашивает свою базу данных и применяет одно или более из сортировки, фильтрации или группировки, определенные клиентом, чтобы получить коллекцию строк. Эта коллекция строк затем обрабатывается для генерации наборов данных для отсылки клиенту.

Неотфильтрованная иерархия является полной иерархической структурой, которая служит в качестве отображения строковых ключей и родительских ключей, по порядку. Чтобы понять, что подразумевается под определением "неотфильтрованный", рассмотрим пример из 3 строк, А, В и С. В является потомком А, а С является потомком В. Может получиться так, что в ходе фильтрации строк В не включится в представление строк и таким образом С не имеет родителя (непосредственного предка) в представлении строк. Неотфильтрованная иерархия является иерархическим отношением между А, В и С. Нет необходимости определять родителя С при прокрутке/разворачивании/сворачивании, но это необходимо при формировании смещения/расформировании смещения/вставке/удалении. Таким образом, во время операций редактирования используется неотфильтрованная иерархия, так как желаемая информация не может быть построена из исследования родительского идентификатора записи каждой строки представления.

Неотфильтрованная иерархическая структура посылается клиенту только во время редактирования и используется для ответа на вопросы о строках при необходимости на клиенте, включая структурный уровень строки независимо от того, является ли строка родителем, какие строки являются дочерними строками, и какая строка является предыдущей одноуровневой строкой для этой строки, для поддержки формирования смещения/расформирования смещения. В режиме редактирования иерархическую структуру посылают клиенту без постраничного разбиения. Вся иерархическая структура появляется для поддержки операций, которые влияют на иерархию (например, формирование смещения/расформирование смещения, вставка или удаление строк).

Строки являются постранично разбитым набором строковых ключей и связанных данных (в случае проектов, это может быть коллекцией задач относительно идентификаторов). Как описано ниже, это включает в себя следующие поля. Структурный уровень: в режиме только для чтения это поле одно включает потребность в неотфильтрованной иерархической структуре; в режиме

редактирования система использует неотфильтрованную иерархию. Родительский строковый ключ: в режиме только для чтения это поле одно исключает потребность в неотфильтрованной иерархии; в режиме редактирования система использует неотфильтрованную иерархическую структуру. Количество потомков: в режиме
5 только для чтения оно вычисляется на сервере и посылается клиенту; в режиме редактирования для вычисления количества потомков используются неотфильтрованная иерархическая структура и представление строк. Другие поля данных могут также использоваться по мере необходимости для системы.

10 Отметим, что весь иерархический набор данных не включается в список выше. Полный список строк не посылается клиенту. Только релевантные строки, требуемые в клиентском наборе представлений относительно фильтрации, сортировки и развернутой/свернутой иерархии, посылаются клиенту.

15 Как показано, расширенное представление 400 включает в себя колонки для идентификатора (ID), Имени, Структурного Уровня, Позиции Представления, Позиции Данных и Количества Потомков. Количество потомков является общим количеством дочерних строк, которые видны по отношению к текущему представлению строк. Другими словами, после применения фильтра это представляет собой количество
20 дочерних, внучатых и т.д. строк, которые можно по возможности послать клиенту. Итоговая строка является строкой в иерархии, у которой есть потомки. В расширенном представлении 400, T1 является итоговой строкой, потому что у нее есть дочерние строки (ПОТОМОК А и ПОТОМОК В). Следует понимать, что оба из представления 400 и представления 402 являются клиентскими представлениями,
25 непосредственно перед и после снимков в примере сворачивания одной строки. Может совпасть так, что представление 400 включает в себя точный набор строк, которые включает в себя представление строк.

30 Как указывалось ранее, позиция представления (ПОЗИЦИЯ ПРЕДСТАВЛЕНИЯ) является порядковой позицией строки в пределах текущего развернутого представления 400 относительно развернутых/свернутых строк, а позиция данных (ПОЗИЦИЯ ДАННЫХ) является порядковой позицией строки в пределах текущего представления 400, которое включают в себя все строки, независимо от того, развернуты строки или нет. Структурный уровень определяется согласно уровню
35 (например, 1, 2, 3...). Здесь, имя T1 является крайним левым структурным уровнем для 1, ПОТОМОК А назначается на структурный уровень 2, ПОТОМОК С дополнительно назначается на структурный уровень 3 и так далее. В развернутом представлении 400 восемь строк; таким образом, номера 1-8 позиций представления
40 назначаются соответствующим ID 1-8. Номер позиции данных назначается строке и следует за строкой, будь она развернута или свернута. Отметим, как сворачивание T1 заставило позиции представления измениться, но позиции данных для строк остаются неизменными.

45 Количество потомков для строки с ID 1 равно двум из-за того, что строка с ID 1 имеет две дочерних строки (ID 2 и ID 3). У дочерних строк (ID 2 и ID 3) нет дочерних строк; таким образом, количество потомков равно нулю для каждой из этих строк (ID 2 и ID 3).

50 Индекс 404 конвертера позиции и список 406 свернутых наборов поддерживаются на клиенте для эффективности. Индекс 404 конвертера позиции является упорядоченным индексом строк, который используется для отслеживания развернутого/свернутого состояния иерархии и трансляции между позицией представления и позицией данных. Индекс 404 конвертера позиции включает в себя

поле строкового ключа, которое является GUID (глобально уникальный ID), который идентифицирует строку в базе данных, поле позиции представления, которое является позицией этой строки от вершины представления относительно любых развернутых/свернутых строк, и позицию данных, которая является абсолютной позицией этой строки от вершины представления (игнорируя свернутые строки). Индекс 404 конвертера позиции обеспечивает возможность быстрого преобразования между позицией представления и позицией данных так, чтобы только строки, которые находятся в клиентском наборе представлений, были получены от сервера, и чтобы строки помещались в правильном визуальном порядке. Эта структура данных создается во время выполнения (рабочего цикла) на клиенте и добавляется, когда строки сворачиваются. Начальное состояние этой таблицы пусто, так как иерархия развернута по умолчанию.

Список 406 свернутых наборов является списком свернутых строк, который позволяет гриду знать, будет ли строка свернута или развернута. Этот список 406 первоначально пуст, но поскольку строки разворачиваются/сворачиваются на клиенте, этот список обновляется.

В этом примере соответствующие значения в индексе 404 конвертера позиции и списке 406 свернутых наборов отслеживают развернутое представление 400 таким образом, что строковые ключи (ID) равны 1-8, позиция представления является 1-8, и позиция данных является 1-8. Поскольку нет никаких свернутых строк в развернутом (полностью) представлении 400, список свернутых наборов пуст.

Продолжая со свернутым представлением 402, строка с ID 1 теперь свернута (как указано символом "+"). Соответствующие значения в индексе 408 конвертера позиции и списке 410 свернутых наборов отслеживают свернутое представление 402.

Сворачивание дочерних строк (ID 2 и ID 3) приводит значения конвертера строковых ключей (ID) к 1-8, позицию представления строковых ключей ID 2 и 3 к отображению строки с ID 1, в которую сворачиваются дочерние строки, а также отображению других позиций представления для оставшихся строковых ключей ID к позиции текущего представления. Позиция данных, как и прежде, является 1-8. Поскольку теперь существуют свернутые строки в представлении 402, список 410 свернутых наборов причисляет строку с ID 1 к свернутым строкам.

Некоторые или все эти коллекции посылаются клиенту. В режиме считывания можно послать представление строк и/или постранично разбитый поднабор строк, включая три дополнительных поля для определения его иерархического отношения с другими строками. В режиме редактирования неотфильтрованная иерархическая структура, представление строк и/или постранично разбитый поднабор строк (не включая три дополнительных поля для определения его иерархического отношения с другими строками - поскольку информация, которую обеспечивают эти поля, может быть получена из неотфильтрованной иерархической структуры).

Клиент может затем решить, какие строки запрашивать с сервера для заполнения клиентского набора представлений в последующих событиях режима считывания и режима редактирования. В режиме считывания, относящемся к сворачиванию, свернутый набор обновляется для включения указанного строкового ключа и затем конвертер позиции обновления для включения новых позиций представления. В режиме считывания, относящемся к разворачиванию, развернутая строка удаляется из свернутого набора, что сопровождается обновлением конвертера позиции для включения новых позиций представления. Клиент затем запрашивает строки, которых у клиента уже нет, у клиентского набора представлений. Когда узел сворачивается, и

узел включает в себя одного или более свернутых потомков, эти записи списка в свернутом наборе присоединяются к записи вновь свернутой строки в свернутом наборе. В режиме считывания, относящемся к прокрутке (вниз/вверх), грид использует запрошенную позицию в клиентском наборе представлений для запроса постранично разбитого набора строк у клиента.

В режиме редактирования, относящемся к вставке, неотфильтрованная иерархия обновляется для включения новой строки в указанную позицию, гарантируя, что новая строка имеет надлежащий родительский ID и порядок среди своих одноуровневых строк. Конвертер позиции обновляется, и все последующие позиции представления увеличиваются на один. Грид затем сохраняет изменения на сервер. Сервер не должен отсылать обновление неотфильтрованной иерархии потому, что клиентская версия совпадает с тем, что сохранено на сервере как неотфильтрованная иерархия, включая дополнение.

В режиме редактирования, относящемся к удалению, неотфильтрованная иерархия обновляется для удаления указанной строки. Конвертер позиции обновляется, и все последующие позиции представления уменьшаются на один. Как и выше, неотфильтрованная иерархическая структура на клиенте правильна. В режиме редактирования, относящемся к формированию смещения, поскольку неотфильтрованная иерархическая структура является деревом, грид ищет указанную строку и перемещает строку в правильное местоположение в дереве. Позиция представления не изменяется, таким образом нет никакой необходимости обновлять конвертер позиции. Отметим, что количество потомков оценивается на основе "по мере необходимости"; грид запрашивает, чтобы строка посчитала своих потомков, и затем кэширует данные, чтобы в следующий раз, когда информация запрашивается, поиск можно было быстро осуществить. В режиме редактирования, относящемся к расформированию смещения, поскольку неотфильтрованная иерархическая структура является деревом, грид ищет указанную строку и перемещает строку в правильное местоположение в дереве.

Группировка является особым случаем иерархии. Для группировки сервер сортирует все данные, генерирует искусственную иерархию и вставляет фиктивные итоговые строки для заголовков группировки. Новое представление строк посылается клиенту как любое другое представление. В сгруппированном представлении структурный уровень равен количеству условий группировки.

Далее следует серия блок-схем последовательности операций, представляющих примерные методики для выполнения новых аспектов раскрытой архитектуры. В то время как в целях простоты объяснения одна или более методик, показанных в данном документе, например, в форме блок-схемы или блок-схемы последовательности операций, показаны и описаны как последовательность действий, должно быть понятно, что методики не ограничены порядком действий, поскольку некоторые действия согласно им же могут происходить в другом порядке и/или одновременно с другими действиями из тех, что показаны и описаны в данном документе. Например, специалистам будет понятно, что методика может альтернативно быть представлена как последовательность взаимосвязанных состояний или событий, таких как на диаграмме состояний. Кроме того, не все действия, проиллюстрированные в методике, могут быть необходимы для реализации новшества.

Фиг.5 иллюстрирует способ обработки данных. На этапе 500 принимается запрос релевантных данных от клиента, которые будут воспроизведены в иерархическом

клиентском наборе представлений на основании строковой операции клиента. На этапе 502 иерархический набор данных постранично разбивается на релевантные данные на основании запроса. На этапе 504 релевантные данные посылаются клиенту для обновления иерархического клиентского набора представлений. Иерархический набор данных может быть постранично разбит асинхронно на основании запроса.

Кроме того, способ может дополнительно содержать управление отношениями "родитель/потомок" на основании строковой операции. Постраничное разбиение иерархического набора данных может быть реакцией на операцию считывания, которая сворачивает строки иерархического клиентского набора представлений в свернутую строку, разворачивает свернутую строку для показа одной или более новых строк в иерархическом клиентском наборе представлений и добавляет или удаляет строки на основании прокрутки иерархического клиентского набора представлений. В качестве альтернативы постраничное разбиение иерархического набора данных может быть реакцией на операцию редактирования, связанную со вставкой строки, удалением строки, формированием смещения строки или расформированием смещения строки в иерархическом представлении строк.

Способ может содержать создание неотфильтрованной иерархической структуры, которая определяет релевантные данные, относящиеся к родителям строки, структурному уровню строки, дочерним строкам и операциям формирования смещения/расформирования смещения, а также и применение одного или более из фильтрации, сортировки или группировки для приема релевантных данных.

Фиг.6 иллюстрирует способ отслеживания операций представлений на стороне клиента. На этапе 600 клиент посылает запрос серверу, относящийся к редактированию строк в иерархическом клиентском наборе представлений. На этапе 602 неотфильтрованная иерархическая структура принимается с сервера. На этапе 604 клиент отслеживает информацию о строковом ключе, информацию о позиции представления и информацию о позиции данных на основании изменений иерархической структуры, используя индекс конвертера позиции. На этапе 606 клиент отслеживает операции по сворачиванию строки на основании изменений иерархической структуры, используя список свернутых наборов. На этапе 608 клиент обрабатывает индекс конвертера позиции и список свернутых наборов для запроса только релевантных строк для строковой операции из иерархического набора данных.

При использовании в данной заявке, термины "компонент" и "система" предназначены для обозначения связанной с компьютером сущности, т.е. либо аппаратного обеспечения, либо комбинации аппаратного и программного обеспечения, либо программного обеспечения, либо исполняемого программного обеспечения. Например, компонент может быть, но не в качестве ограничения, процессом, исполняемым процессором, процессором, жестким диском, несколькими накопителями (на оптических и/или магнитных носителях), объектом, исполняемым файлом, исполняемым потоком, программой и/или компьютером. В качестве иллюстрации, как приложение, исполняемое на сервере, так и сервер может быть компонентом. Один или более компонентов могут располагаться в рамках процесса и/или исполняемого потока, и компонент может быть расположен на одном компьютере и/или быть распределенным между двумя или более компьютерами. Слово "примерный" может использоваться в данном документе для обозначения чего-либо служащего примером, экземпляром или иллюстрацией. Любой аспект или исполнение, описанные в данном документе как "примерные", не должны обязательно рассматриваться как предпочтительные или преимущественные в сравнении с другими

аспектами или исполнениями.

Обратимся теперь к фиг.7. На ней проиллюстрирована блок-схема вычислительной системы 700, способной выполнять процессы постраничного разбиения и представления в соответствии с раскрытой архитектурой. Чтобы обеспечить
5 дополнительный контекст для различных аспектов этого, фиг.7 и последующее обсуждение предназначены для обеспечения краткого, общего описания подходящей вычислительной системы 700, в которой эти различные аспекты могут быть реализованы. Хотя описание выше приведено в общем контексте исполняемых
10 компьютером инструкций, которые могут исполняться на одном или более компьютерах, специалисты признают, что новый вариант осуществления также может быть реализован в комбинации с другими программными модулями и/или как комбинация аппаратного и программного обеспечения.

В общем случае, программные модули включают в себя процедуры, программы,
15 компоненты, структуры данных и т.д., которые выполняют конкретные задачи или реализуют конкретные абстрактные типы данных. Кроме того, специалистам будет понятно, что отвечающие изобретению способы могут быть осуществлены с другими конфигурациями компьютерной системы, включая однопроцессорные или
20 многопроцессорные компьютерные системы, миникомпьютеры, мейнфрейм-компьютеры так же, как и персональные компьютеры, карманные вычислительные устройства, основанную на микропроцессорах или программируемую бытовую электронику и т.п., каждое из которых может быть функционально соединенным с одним или более ассоциированных устройств.

Проиллюстрированные аспекты могут также быть осуществлены в распределенном вычислительном окружении, где определенные задачи выполняются удаленными
устройствами обработки, которые связаны через систему связи. В распределенном вычислительном окружении программные модули могут быть расположены как в
30 локальных, так и в удаленных устройствах хранения данных.

Компьютер обычно включает в себя множество считываемых компьютером носителей. Считываемые компьютером носители могут быть любыми доступными носителями, к которым может осуществить доступ компьютер, и включать в себя
35 энергозависимые и энергонезависимые носители, съемные и несъемные носители. В качестве примера, но не ограничения, считываемые компьютером носители могут содержать компьютерные носители данных и среды связи. Компьютерные носители данных включают в себя энергозависимые и энергонезависимые, съемные и несъемные носители, осуществленные любым способом или технологией для хранения
40 информации, такой как считываемые компьютером инструкции, структуры данных, программные модули или другие данные. Компьютерные носители данных включают в себя, но не в качестве ограничения, RAM, ROM, EEPROM, флэш-память или другую технологию памяти, CD-ROM, цифровой видеодиск (DVD) или другое оптическое дисковое хранилище, магнитные кассеты, магнитную ленту, магнитное дисковое
45 хранилище или другие магнитные устройства хранения, или любой другой носитель, который может использоваться для хранения желаемой информации и к которому может осуществить доступ компьютер.

Со ссылкой снова на фиг.7 описывается примерная вычислительная система 700 для реализации различных аспектов, которая включает в себя компьютер 702, имеющий
50 блок 704 обработки, системную память 706 и системную шину 708. Системная шина 708 обеспечивает интерфейс для системных компонентов, включая, но не в качестве ограничения, системную память 706, с блоком 704 обработки. Блок 704

обработки может быть любым из различных коммерчески доступных процессоров. Двойные микропроцессоры и другие многопроцессорные архитектуры могут также использоваться в качестве блока 704 обработки.

5 Системная шина 708 может быть любой из нескольких типов шинных структур, которая может быть дополнительно взаимосвязана с шиной памяти (непосредственно или через контроллер памяти), периферийной шиной и локальной шиной, с использованием любой из множества коммерчески доступных шинных архитектур. Системная память 706 может включать в себя энергонезависимую память (NON-VOL)
10 710 и/или энергозависимую память 712 (например, оперативную память (RAM)). Базовая система ввода/вывода (BIOS) может быть сохранена в энергонезависимой памяти 710 (например, ROM, EPROM, EEPROM и т.д.), причем BIOS представляет собой базовые процедуры, которые помогают переносить информацию между
15 элементами в пределах компьютера 702, как, например, во время запуска. Энергозависимая память 712 может также включать в себя высокоскоростную RAM, такую как статическая RAM для кэширования данных.

Компьютер 702 дополнительно включает в себя внутренний накопитель 714 на жестких дисках (HDD) (например, EIDE, SATA), причем внутренний HDD 714 может
20 также быть сконфигурирован для внешнего использования в подходящем корпусе, накопитель 716 на магнитных гибких дисках (FDD) (например, для считывания или записи на съемную дискету 718) и накопитель 720 на оптических дисках (например, для считывания диска 722 CD-ROM или для считывания с или записи на другой оптический
25 носитель высокой емкости, такой как DVD). HDD 714, FDD 716 и накопитель 720 на оптических дисках могут быть соединены с системной шиной 708 через интерфейс 724 HDD, интерфейс 726 FDD и интерфейс 728 накопителя на оптических дисках соответственно. Интерфейс 724 HDD для реализаций с внешним накопителем может
30 включать в себя по меньшей мере одну или обе из технологий интерфейсов Универсальной Последовательной шины (USB) и IEEE 1394.

Накопители и связанные с ними считываемые компьютером носители обеспечивают энергонезависимое хранилище данных, структур данных, исполняемых компьютером
инструкций и т.д. Для компьютера 702, накопители и носители обеспечивают хранилище любых данных в подходящем цифровом формате. Хотя описание
35 считываемых компьютером носителей выше относится к HDD, съемной магнитной дискете (например, FDD) и съемным оптическим носителям, таким как CD или DVD, специалистам должно быть понятно, что другие типы носителей, которые являются считываемыми компьютером, такие как zip-накопители, магнитные кассеты, карты
40 флэш-памяти, картриджи и т.п., могут также использоваться в примерном рабочем окружении, и дополнительно, что любые такие носители могут содержать исполняемые компьютером инструкции для выполнения новых способов раскрытой архитектуры.

Некоторые программные модули могут быть сохранены на накопителях и
45 энергозависимой памяти 712, включая операционную систему 730, одну или более прикладных программ 732, другие программные модули 734 и программные данные 736. При использовании в качестве клиентской вычислительной системы они могут включать в себя одну или более прикладных программ 732, другие
50 программные модули 734, и программные данные 736 могут включать в себя иерархический клиентский набор 110 представлений, строки 108, строковую операцию 106, операцию 114 изменения строк, клиент 204, представление 210 строк, клиентское представление 212, строковые операции 300, индекс 302 конвертера

позиции, список 304 свернутых наборов, развернутое представление 400, свернутое представление 402, индексы конвертера позиции (404 и 408), списки свернутых наборов (406 и 410) и способ с фиг.6, например. При использовании компьютера 702 как серверной системы одна или более прикладных программ 732, другие программные модули 734 и программные данные 736 могут включать в себя компонент 102 постраничного разбиения, компонент 112 обновления, иерархический набор 104 данных, сервер 202, представление 206 набора данных, представление 208 строк и способ с фиг.5, например.

Вся или части операционной системы, приложений, модулей и/или данных могут также быть кэшированы в энергозависимой памяти 712. Должно быть понятно, что раскрытая архитектура может быть реализована с различными коммерчески доступными операционными системами или комбинациями операционных систем.

Пользователь может вводить команды и информацию в компьютер 702 через одно или более проводных/беспроводных устройств ввода, например клавиатуру 738 и указательное устройство, такое как мышь 740. Другие устройства ввода (не показаны) могут включать в себя микрофон, инфракрасный пульт дистанционного управления, джойстик, геймпад, перо, сенсорный экран и т.п. Эти и другие устройства ввода часто соединяются с блоком 704 обработки через интерфейс 742 устройств ввода, который соединен с системной шиной 708, но может быть соединен с другими интерфейсами, такими как параллельный порт, последовательный порт IEEE 1394, игровой порт, порт USB, интерфейс IR и т.д.

Монитор 744 или другой тип дисплейного устройства также соединен с системной шиной 708 через интерфейс, такой как видеоадаптер 746. В дополнение к монитору 744 компьютер обычно включает в себя другие периферийные устройства вывода (не показаны), такие как громкоговорители, принтеры и т.д.

Компьютер 702 может работать в сетевом окружении, используя логические соединения через проводную и/или беспроводную связи с одним или более удаленными компьютерами, такими как удаленный компьютер(ы) 748. Удаленный компьютер(ы) 748 может быть автоматизированным рабочим местом, серверным компьютером, маршрутизатором, персональным компьютером, портативным компьютером, основанным на микропроцессоре развлекательным устройством, одноранговым устройством или другим общим узлом сети и обычно включает в себя многие или все из элементов, описанных в отношении компьютера 702, хотя в целях краткости только проиллюстрирована память/устройство хранения 750. Изображенные логические соединения включают в себя проводные/беспроводные соединения с локальной сетью (LAN) 752 и/или большими сетями, например глобальной сетью (WAN) 754. Такие сетевые окружения LAN и WAN являются обычными в офисах и компаниях и обеспечивают компьютерные сети в масштабах всего предприятия, такие как интранет-сети, все из которых могут соединяться с сетью глобальной связи, например Интернет.

При использовании в сетевом окружении LAN компьютер 702 соединен с LAN 752 через сетевой интерфейс или адаптер 756 проводной и/или беспроводной связи. Адаптер 756 может обеспечить проводную и/или беспроводную связь с LAN 752, которая может также включать в себя точку доступа, расположенную в ней, для осуществления связи с функциональными средствами беспроводной связи адаптера 756.

При использовании в сетевом окружении WAN, компьютер 702 может включать в себя модем 758, или может быть соединен с сервером связи WAN 754, или имеет другие

средства для установления связи через WAN 754, как, например, с помощью Интернет. Модем 758, который может быть внутренним или внешним и проводным и/или беспроводным устройством, соединен с системной шиной 708 через интерфейс 742 устройства ввода. В сетевом окружении программные модули, изображенные в отношении компьютера 702 или его частей, могут быть сохранены в удаленной памяти/устройстве хранения 750. Должно быть понятно, что показанные сетевые соединения являются примерными, и могут быть использованы другие средства установления линии связи между компьютерами.

Компьютер 702 способен осуществлять связь с проводными и беспроводными устройствами или субъектами, используя семейство стандартов IEEE 802, такими как беспроводные устройства, функционально осуществляющие беспроводную связь (например, эфирные методики модуляции IEEE 802.11) с, например, принтером, сканером, настольным и/или портативным компьютером, персональным цифровым помощником (PDA), спутником связи, любым элементом оборудования или местоположением, связанными с беспроводным образом детектируемой меткой (например, киоск, газетный киоск, уборная) и телефоном. Это включает в себя по меньшей мере Wi-Fi (или Беспроводная Точность), WiMax и беспроводные технологии Bluetoothtm. Таким образом, связь может быть predetermined структуры, как с традиционной сетью, или просто одноранговой самоорганизующейся связью между по меньшей мере двумя устройствами. Сети Wi-Fi используют радиотехнологии, называемые IEEE 802.11x (a, b, g и т.д.), для обеспечения безопасной, надежной, быстрой беспроводной возможности соединения. Сеть Wi-Fi может использоваться для соединения компьютеров друг с другом, с Интернет и с проводными сетями (которые используют связанные с IEEE 802.3 носители и функции).

Обратимся теперь к фиг.8. На ней проиллюстрирована схематическая блок-схема примерного вычислительного окружения 800 для выполнения процессов представления и постраничного разбиения. Окружение 800 включает в себя один или более клиентов 802. Клиент(ы) 802 может являться аппаратным обеспечением и/или программным обеспечением (например, потоки, процессы, вычислительные устройства). Клиент(ы) 802 может размещать cookies и/или ассоциированную контекстную информацию, например.

Окружение 800 также включает в себя один или более сервер(ов) 804. Сервер(ы) 804 может также быть аппаратным обеспечением и/или программным обеспечением (например, потоки, процессы, вычислительные устройства). Серверы 804 могут размещать потоки для выполнения преобразований, используя упомянутую архитектуру, например. Одна возможная связь между клиентом 802 и сервером 804 может осуществляться в форме пакетов данных, приспособленных для передачи между двумя или более компьютерными процессами. Пакет данных может включать в себя cookies и/или ассоциированную контекстную информацию, например.

Окружение 800 включает в себя инфраструктуру 806 связи (например, сеть глобальной связи, такую как Интернет), которая может быть использована для обеспечения связи между клиентом(ами) 802 и сервером(ами) 804.

Связи могут быть обеспечены посредством проводов (включая оптоволокно) и/или беспроводной технологии. Клиент(ы) 802 является функционально связанным с одним или более хранилищем(ами) 808 данных клиента, которые могут быть использованы для хранения информации, являющейся локальной по отношению к клиенту(ам) 802 (например, cookies и/или ассоциированной контекстной информации). Точно так же сервер(ы) 804 является функционально связанным с одним или более

хранилищем(ами) 810 данных сервера, которые могут быть использованы для хранения информации, являющейся локальной по отношению к серверам 804.

Клиент(ы) 802 может включать в себя клиент 204 и способ с фиг.6, а сервер(ы) может включать в себя сервер 202 и способ с фиг.5.

5 То, что было описано выше, включает в себя примеры раскрытой архитектуры. Конечно, невозможно описать каждую мыслимую комбинацию компонентов и/или методик, но средний специалист сможет признать, что возможны многие дополнительные комбинации и перестановки. Соответственно, новая архитектура
10 подразумевается охватывающей все такие изменения, модификации и вариации, которые находятся в пределах объема прилагаемой формулы изобретения. Кроме того, в той степени, в которой термин "включающий в себя" используется либо в подробном описании, либо в формуле, этот термин подразумевается включающим
15 подобно интерпретации термина "содержащий" при использовании термина "содержащий" как связующего слова в формуле изобретения.

Формула изобретения

1. Компьютерно-реализованная система обработки данных, содержащая:
20 компонент постраничного разбиения для постраничного разбиения иерархического набора данных, содержащего таблицу, в которой имеются строки, для получения строк, релевантных строковой операции в отношении одной или более строк клиентского набора представлений, тем самым сводя количество обрабатываемых
25 данных к этим релевантным строкам, при этом при постраничном разбиении получают поднабор строк посредством операций представления, включая сортировку, фильтрацию или группирование, причем клиентский набор представлений представляет собой то, что пользователь может физически видеть на клиенте в данный момент;

30 компонент обновления для обновления клиентского набора представлений так, чтобы он включал в себя упомянутые релевантные строки, в качестве реакции на операцию изменения строки, содержащую строковую операцию, имеющую характер редактирования, причем обновление клиентского набора представлений основано на
35 строковых ключах, связанных с постранично разбитым иерархическим набором данных; и

процессор, который исполняет машиноисполняемые инструкции, связанные с по меньшей мере одним из компонента постраничного разбиения и компонента обновления.

40 2. Система по п.1, в которой компонент постраничного разбиения управляет отношениями "родитель/потомок" строк в клиентском наборе представлений, причем строки являются релевантными строковой операции.

3. Система по п.1, в которой компонент постраничного разбиения создает индекс структуры иерархии, определяемый упорядоченным отображением строковых ключей
45 и родительских ключей, причем индекс посылается клиенту в качестве реакции на строковую операцию клиентского набора представлений.

4. Система по п.1, в которой строковая операция является операцией считывания, которая сворачивает строки иерархического представления в свернутую строку,
50 разворачивает свернутую строку для показа одной или более новых строк в иерархическом представлении и добавляет или удаляет строки на основе прокрутки клиентского набора представлений.

5. Система по п.1, в которой строковая операция является операцией

редактирования, связанной со вставкой строки, удалением строки, формированием смещения строки или расформированием смещения строки в клиентском наборе представлений.

5 6. Система по п.1, в которой компонент постраничного разбиения постранично разбивает иерархический набор данных на строки, релевантные строковой операции, на основе фильтрования, сортировки, разворачивания строк и сворачивания строк.

7. Система по п.1, в которой компонент обновления обновляет неотфильтрованную иерархию на основе строковых операций, связанных с режимом редактирования.

10 8. Система по п.1, в которой компонент постраничного разбиения постранично разбивает иерархический набор данных асинхронно на основе операции изменения строки.

9. Компьютерно-реализованная система обработки данных, содержащая:
15 компонент постраничного разбиения из состава сервера для получения поднабора релевантных строк иерархического набора данных, определяемого таблицей, включающей в себя строки, в качестве реакции на строковую операцию для получения релевантных данных, подлежащих воспроизведению в клиентском наборе представлений клиента, при этом клиентский набор представлений содержит набор
20 строк, видимых на экране клиента без прокрутки;

компонент обновления из состава сервера для обновления клиентского набора представлений так, чтобы он включал в себя упомянутые релевантные строки, согласно операции изменения строки, содержащей строковую операцию, имеющую
25 характер редактирования, причем упомянутый поднабор включает в себя только строки, релевантные операции изменения строки и

процессор, который исполняет машиноисполняемые инструкции, связанные с по меньшей мере одним из компонента постраничного разбиения и компонента
обновления.

30 10. Система по п.9, в которой компонент постраничного разбиения управляет отношениями "родитель/потомок" строк в клиентском наборе представлений.

11. Система по п.9, в которой компонент постраничного разбиения создает индекс структуры иерархии, определяемый упорядоченным отображением строковых ключей и родительских ключей, и посылает индекс клиенту с обновленным клиентским
35 набором представлений на основе операции изменения строки.

12. Система по п.9, в которой строковая операция является операцией считывания, причем операция считывания связана со сворачиванием строк клиентского набора представлений в свернутую строку, разворачиванием свернутой строки для показа
40 одной или более новых строк в клиентском наборе представлений или добавлением и удалением строк на основе прокрутки клиентского набора представлений.

13. Система по п.9, в которой строковая операция является операцией редактирования, связанной со вставкой строки, удалением строки, формированием смещения строки или расформированием смещения строки в клиентском наборе
45 представлений.

14. Компьютерно-реализуемый способ обработки данных, содержащий этапы, на которых:

предоставляют иерархический набор данных, определяемый таблицей, содержащей
50 строки;

принимают от клиента запрос релевантных данных, подлежащих воспроизведению в клиентском наборе представлений на основе строковой операции клиента, причем клиентский набор представлений содержит набор строк, видимых на экране клиента

без прокрутки;

получают поднабор релевантных строк иерархического набора данных в качестве релевантных данных в ответ на упомянутый запрос путем выполнения операций представления, включая сортировку, фильтрование или группирование; и

5 посылают эти релевантные строки клиенту для обновления клиентского набора представлений.

15. Способ по п.14, дополнительно содержащий этап, на котором выполняют управление отношениями "родитель/потомок" на основе строковой операции.

10 16. Способ по п.14, дополнительно содержащий этап, на котором выполняют постраничное разбиение иерархического набора данных в качестве реакции на операцию считывания, которая сворачивает строки клиентского набора представлений в свернутую строку, разворачивает свернутую строку для показа одной или более дочерних строк в клиентском наборе представлений и показывает/скрывает

15 строки на основе прокрутки клиентского набора представлений.
17. Способ по п.14, дополнительно содержащий этап, на котором получают поднабор иерархического набора данных в качестве реакции на операцию редактирования, связанную со вставкой строки, удалением строки, формированием

20 смещения строки или расформированием смещения строки в клиентском наборе представлений.
18. Способ по п.14, дополнительно содержащий этап, на котором применяют одно или более из фильтрования, сортировки или группировки для приема релевантных

25 данных.
19. Способ по п.14, дополнительно содержащий этап, на котором создают неотфильтрованную иерархическую структуру, которая определяет релевантные данные, относящиеся к родителям строки, структурному уровню строки, дочерним

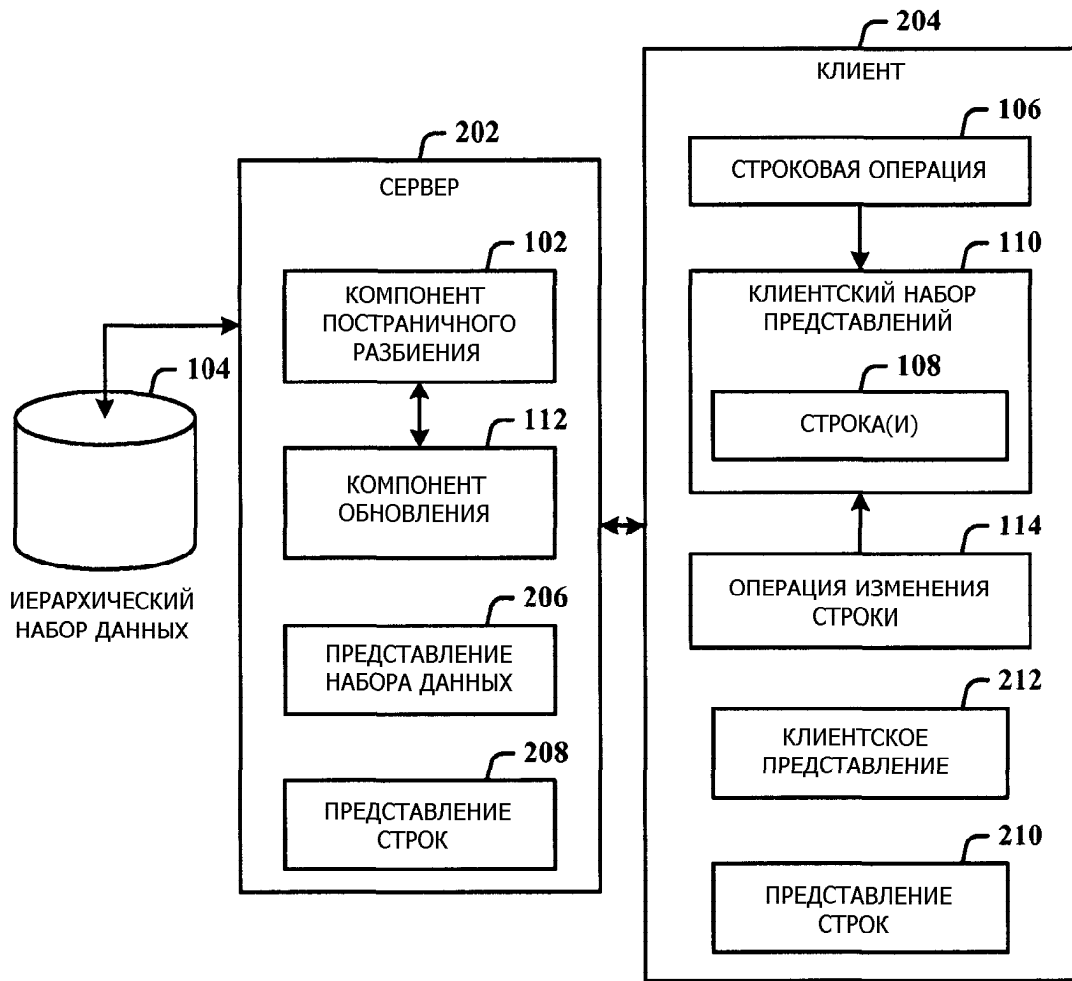
30 строкам и операциям формирования смещения/расформирования смещения.
20. Способ по п.14, в котором поднабор иерархического набора данных получают асинхронно на основе упомянутого запроса.

35

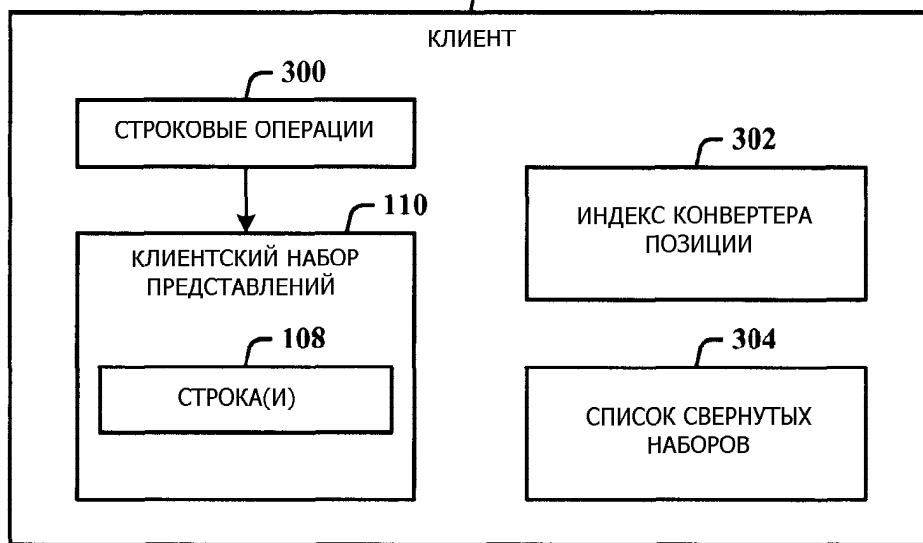
40

45

50



Фиг. 2



Фиг. 3

400 ↗

ID	ИМЯ	СТРУКТУРНЫЙ УРОВЕНЬ	ПОЗИЦИЯ ПРЕДСТАВЛЕНИЯ	ПОЗИЦИЯ ДАННЫХ	КОЛИЧЕСТВО ПОТОМКОВ
1	<input type="checkbox"/> T1	1	1	1	2
2	ПОТОМОК А	2	2	2	0
3	ПОТОМОК В	2	3	3	0
4	T2	1	4	4	0
5	<input type="checkbox"/> T3	1	5	5	3
6	<input type="checkbox"/> ПОТОМОК С	2	6	6	1
7	ПОТОМОК С1	3	7	7	0
8	ПОТОМОК D	2	8	8	0

404

ИНДЕКС КОНВЕРТЕРА ПОЗИЦИИ		
СТРОКОВЫЙ КЛЮЧ	ПОЗИЦИЯ ПРЕДСТАВЛЕНИЯ	ПОЗИЦИЯ ДАННЫХ
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8

406

СПИСОК СВЕРНУТЫХ НАБОРОВ
СТРОКОВЫЙ КЛЮЧ

402 ↗

ID	ИМЯ	СТРУКТУРНЫЙ УРОВЕНЬ	ПОЗИЦИЯ ПРЕДСТАВЛЕНИЯ	ПОЗИЦИЯ ДАННЫХ	КОЛИЧЕСТВО ПОТОМКОВ
1	<input checked="" type="checkbox"/> T1	1	1	1	2
4	T2	1	2	4	0
5	<input type="checkbox"/> T3	1	3	5	3
6	<input type="checkbox"/> ПОТОМОК С	2	4	6	1
7	ПОТОМОК С1	3	5	7	0
8	ПОТОМОК D	2	6	8	0

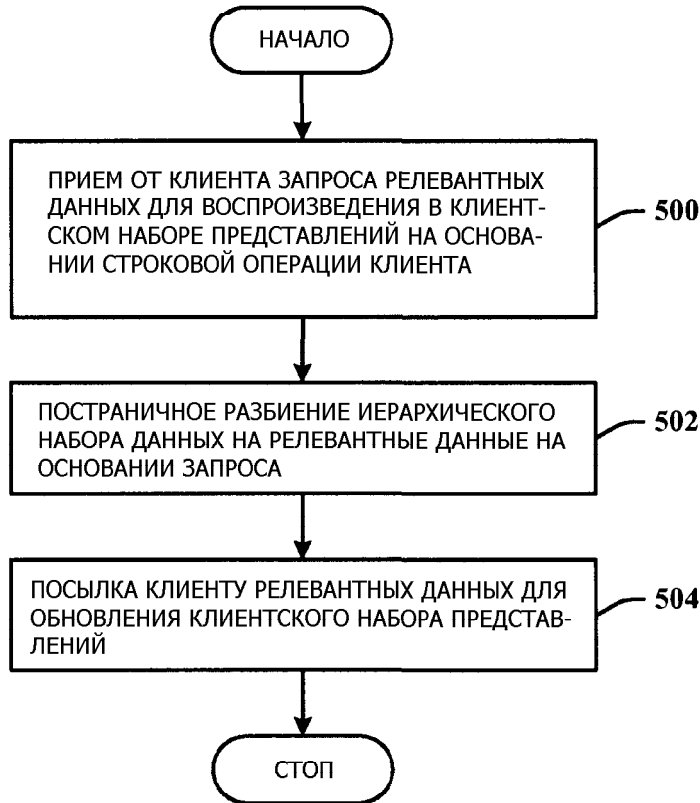
408

ИНДЕКС КОНВЕРТЕРА ПОЗИЦИИ		
СТРОКОВЫЙ КЛЮЧ	ПОЗИЦИЯ ПРЕДСТАВЛЕНИЯ	ПОЗИЦИЯ ДАННЫХ
1	1	1
2	1	2
3	1	3
4	2	4
5	3	5
6	4	6
7	5	7
8	6	8

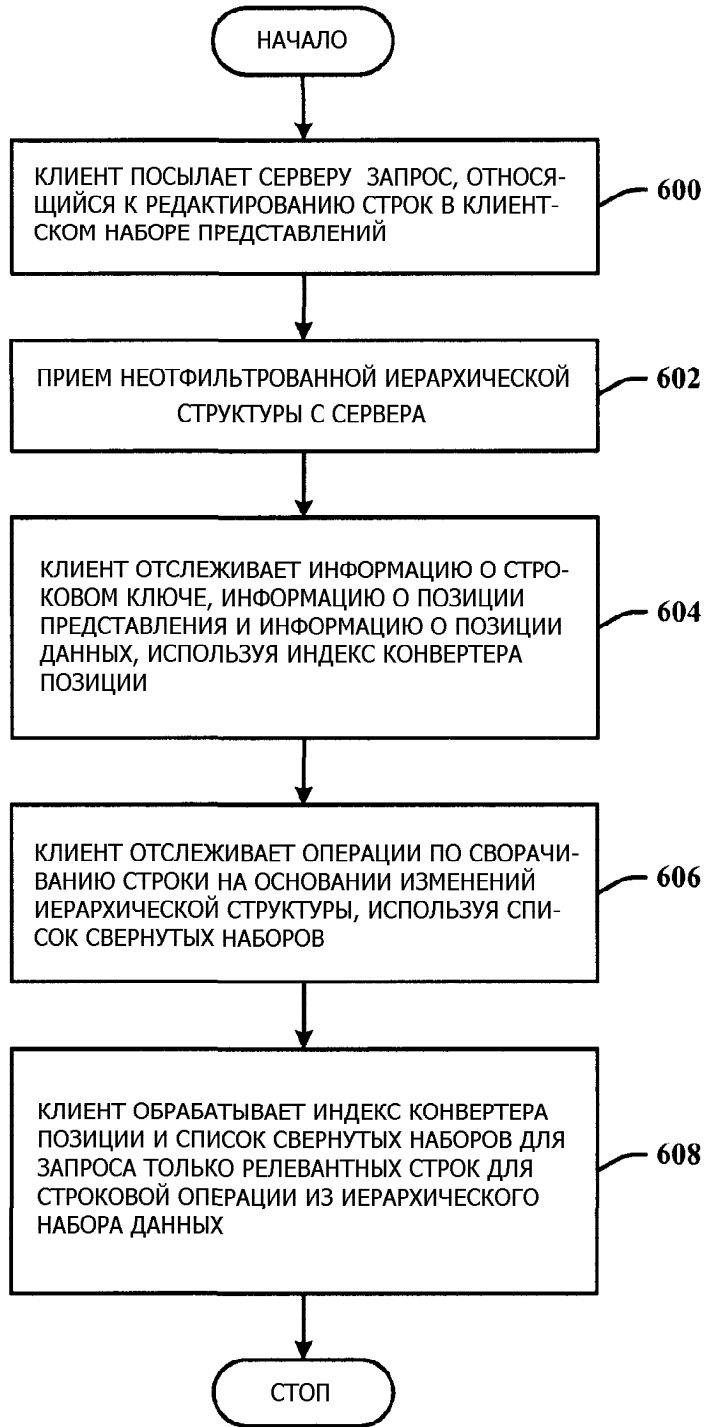
410

СПИСОК СВЕРНУТЫХ НАБОРОВ
СТРОКОВЫЙ КЛЮЧ
1

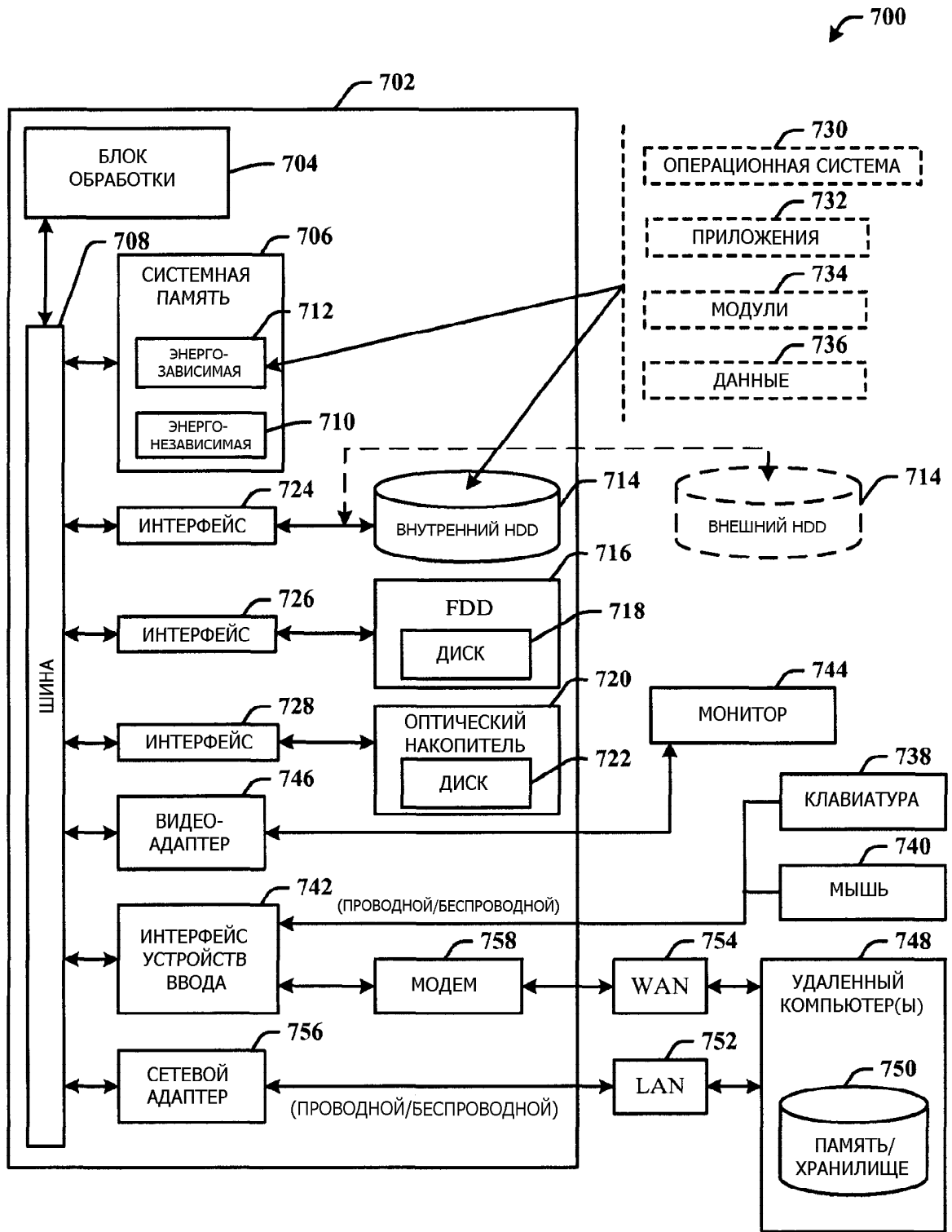
Фиг. 4



Фиг. 5

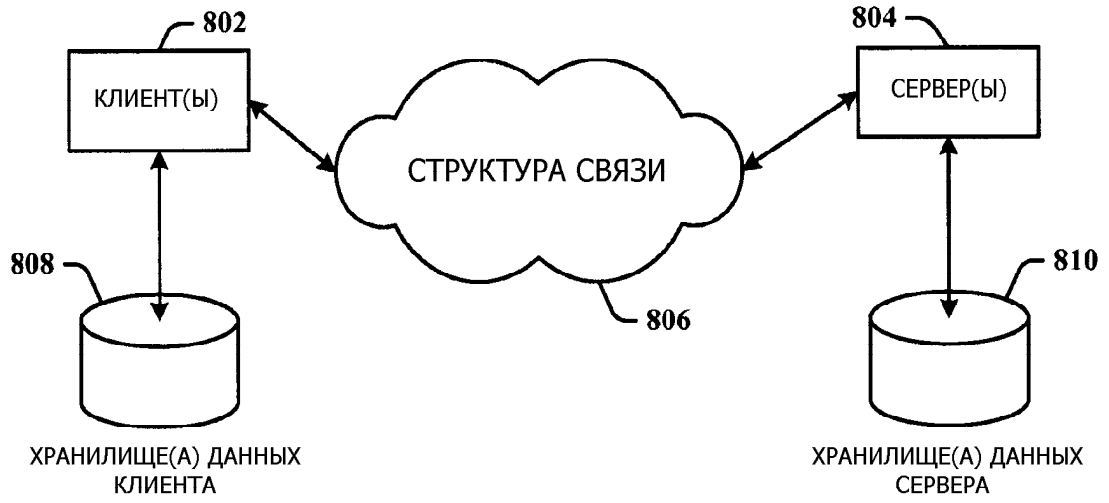


ФИГ. 6



Фиг. 7

800



ФИГ. 8