(19) Евразийское патентное ведомство

(21) 201892256 (13) A1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ

(43) Дата публикации заявки
2020.05.29

(22) Дата подачи заявки
2018.11.02

(51) Int. Cl. ***G06F 16/11*** (2006.01)
G06F 12/08 (2006.01)

(54) СПОСОБ И СИСТЕМА КОМПЛЕКСНОГО УПРАВЛЕНИЯ БОЛЬШИМИ ДАННЫМИ

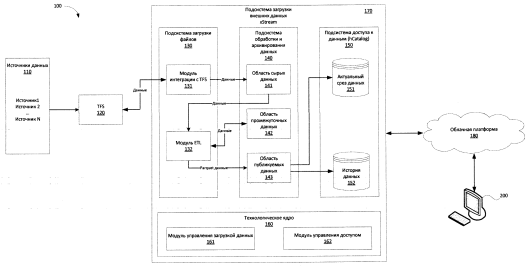
(31) **2018137863**
(32) **2018.10.26**
(33) **RU**

(71) Заявитель:
ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК) (RU)

(72) Изобретатель:
Шарануца Виталий Алексеевич, Булавин Алексей Александрович (RU)

(74) Представитель:
Герасин Б.В. (RU)

(57) Заявленное изобретение относится к управлению большими объемами данных, в частности к системе и способу для их обработки и унифицированного хранения. Достигаемый технический результат совпадает с решаемой технической проблемой и заключается в повышении эффективности хранения больших объемов данных за счет применения распределенной архитектуры хранения с обеспечением обработки входных данных с целью их унификации. Заявленное изобретение осуществляется с помощью системы комплексного управления большими данными (Big Data), содержащей подсистему транспортировки и проверки входных данных (далее - TFS), подсистему загрузки внешних данных (далее - xStream), функционирующую на основании стандарта описания данных, в которой TFS осуществляет прием, проверку и передачу в xStream данных, поступающих от источников данных, причем TFS принимает данные в архивированном виде и осуществляет передачу данных по транспортному протоколу; xStream содержит подсистему загрузки файлов, подсистему обработки и архивирования данных, подсистему доступа к данным (далее - hCatalog), модуль управления загрузкой данных и модуль управления доступом, причем в xStream подсистема загрузки файлов осуществляет опрос TFS для передачи данных, поступивших от источников, в подсистему обработки и архивирования данных, причем источники данных предварительно регистрируются в подсистеме загрузки данных; подсистема загрузки данных выполняет первичное копирование файлов, поступающих от TFS, в подсистему обработки и архивирования; подсистема обработки и архивирования содержит области хранения данных, которые осуществляют структурированное хранение первичных данных, промежуточных данных и публикуемых данных; в область хранения первичных данных передаются файлы из подсистемы загрузки данных, которые соответствуют установленным параметрам архивирования и хранятся в первоначально заархивированном виде; подсистема обработки и архивирования выполнена с возможностью передачи в область промежуточных данных разархивированных первичных данных, преобразованных в формат parquet для дальнейших преобразований; в область публикуемых данных передаются подготовленные, преобразованные, унифицированные данные, приведенные к стандарту xStream, и из унифицированных данных области публикуемых данных формируется структурированный каталог для доступа к упомянутым данным посредством hCatalog.



201892256 A1

201892256 A1

МПК: G06F 17/00; G06F 17/30

СПОСОБ И СИСТЕМА КОМПЛЕКСНОГО УПРАВЛЕНИЯ БОЛЬШИМИ ДАННЫМИ

ОБЛАСТЬ ТЕХНИКИ

[0001] Заявленное решение относится к управлению большими объемами данных, в частности, к системе и способу для их обработки и унифицированного хранения.

УРОВЕНЬ ТЕХНИКИ

[0002] Из уровня техники известны различные решения по организации систем хранения данных на основе распределенных файловых систем (HDFS/ Hadoop File System), которые, в частности, применяются для работы с большими данными.

[0003] Под термином «большие данные» (Big Data) понимается огромный объем данных, поступающих в систему хранения. Данные, как правило, поступают из множества источников информации в неструктурированном виде. Big Data включает в себя также технологии их обработки и использования, методы поиска необходимой информации в больших массивах.

[0004] Основная проблема больших данных связана с большим количеством информации, которую необходимо обрабатывать и, соответственно, хранить доверенным образом с минимизацией рисков потери данных. В связи с этим возникает необходимость резервного копирования данных, которая подразумевает организацию объемных структур хранения данных. Однако с увеличением объема информации растут сложности с ее резервированием.

[0005] HDFS, или Распределенная файловая система Hadoop – это основная система хранения данных, используемая приложениями Hadoop. HDFS многократно копирует блоки данных и распределяет эти копии по вычислительным узлам кластера, тем самым обеспечивая высокую надежность и скорость вычислений.

[0006] Из уровня техники известна архитектура хранилища данных для предоставления совместного доступа к данным, хранимым в нем (US 20130262615, 03.10.2013). Хранилище данных может быть реализовано с помощью HDFS и представлять фильтр для поступающей информации для ее первичной обработки при непосредственном сохранении для последующего использования. Система хранения функционирует с помощью формирования копий, поступающих данных из разнородных источников для последующей связки их с помощью метаданных для целей доступа к ним.

[0007] В заявке CN 106407309 (Wei et al., 15.02.2017) описывается механизм кластерного хранилища для получения информации из множества различных источников данных и обеспечения процесса аутентификации между базой данных и источниками данных.

[0008] Известные решения обладает существенными недостатками, которые заключаются в отсутствии обработки поступающих данных с целью их преобразования в унифицированный вид, что не позволяет оперативно обновлять информацию от источников, и снижает эффективность работы хранилища данных. Также, отсутствие процесса унификации данных приводит к низкой отказоустойчивости системы хранения в случае поступления данных в неправильном или неподдерживаемом формате (ошибки транспортного формата, ошибки архивирования, ошибки копирования, ошибки структуры данных, ошибки типов данных).

РАСКРЫТИЕ ИЗОБРЕТЕНИЯ

[0009] Решаемой технической проблемой с помощью заявленного способа и системы является устранение недостатков, присущих известным аналогам, а также создание нового принципа эффективного хранения больших объемов данных.

[0010] Достигаемый технический результат совпадает с решаемой технической проблемой и заключается в повышении эффективности хранения больших объемов данных, за счет применения распределенной архитектуры хранения с обеспечением обработки входных данных с целью их унификации.

[0011] Дополнительным техническим эффектом является повышение отказоустойчивости системы хранения данных, за счет архитектуры системы управления хранением больших данных и обеспечения унификации хранимых данных.

[0012] Заявление решение осуществляется с помощью системы комплексного управления большими данными (Big Data), содержащая подсистему транспортировки и проверки входных данных (далее TFS), подсистему загрузки внешних данных (далее xStream), функционирующую на основании стандарта описания данных, в которой:

TFS осуществляет прием, проверку и передачу в xStream данных, поступающих от источников данных, причем TFS принимает данные в архивированном виде и осуществляет передачу данных по транспортному протоколу;

xStream содержит подсистему загрузки файлов, подсистему обработки и архивирования данных, подсистему доступа к данным (далее hCatalog), модуль управления загрузкой данных и модуль управления доступом, причем в xStream:

подсистема загрузки файлов осуществляет опрос TFS для передачи данных, поступивших от источников, в подсистему обработки и архивирования данных, причем источники данных предварительно регистрируются в подсистеме загрузки данных;
подсистема загрузки данных выполняет первичное копирование файлов, поступающих от TFS, в подсистему обработки и архивирования;
подсистема обработки и архивирования содержит области хранения данных, которые осуществляют структурированное хранение первичных данных, промежуточных данных и публикуемых данных;
в область хранения первичных данных передаются файлы из подсистемы загрузки данных, которые соответствуют установленным параметрам архивирования и хранятся в первоначально заархивированном виде;
подсистема обработки и архивирования выполнена с возможностью передачи в область промежуточных данных разархивированных первичных данных, преобразованных в формат parquet для дальнейших преобразований;
в область публикуемых данных передаются подготовленные, преобразованные, унифицированные данные, приведенные к стандарту xStream, в которых имена таблиц и полей приведены к стандарту xStream, типы данных приведены к стандарту xStream, таблицы дополнены обязательными техническими полями, описывающими поставку данных - датами начала и конца периода актуальности поставки данных, номерами потоков, выполнивших загрузку и преобразование;
и из унифицированных данных области публикуемых данных формируется структурированный каталог для доступа к упомянутым данным посредством hCatalog.

[0013] В одном из частных вариантов реализации системы TFS осуществляет проверку целостности получаемых архивных данных.

[0014] В другом частном варианте реализации системы для зарегистрированных источников в модуле управления загрузкой хранится ID упомянутых источников.

[0015] В другом частном варианте реализации системы модуль управления загрузкой осуществляет управление потоком данных множества источников по соответствующим сохраненным ID.

[0016] В другом частном варианте реализации системы для каждого источника данных в модуле управления загрузкой содержатся параметры загрузки данных.

[0017] В другом частном варианте реализации системы подсистема загрузки данных осуществляет загрузку файлов в подсистему обработки и архивирования на основании маски загрузки файлов.

[0018] В другом частном варианте реализации системы маска загрузки файлов формируется на основании, по меньшей мере, имени загруженного архивного файла.

[0019] В другом частном варианте реализации системы в подсистеме обработки и архивирования в каждой из областей хранения данных формируется каталог для хранения данных соответствующего источника.

[0020] В другом частном варианте реализации системы модуль управления загрузкой проверяет наличие информации по всем источникам в TFS.

[0021] В другом частном варианте реализации системы выполняется полная или инкрементальная загрузка данных из TFS.

[0022] В другом частном варианте реализации системы инкрементальная загрузка выполняется модулем загрузки данных при обнаружении в TFS новых данных отличающихся от поставленных ранее по дате поставки архива файлов.

[0023] В другом частном варианте реализации системы подсистема обработки и архивирования осуществляет обработку parquet файлов для их приведения в соответствие типам Hive-SQL.

[0024] В другом частном варианте реализации системы для файлов при их обработке подсистемой обработки и архивирования выполняется проверка на наличие аналогичных данных, сохраненных ранее.

[0025] В другом частном варианте реализации системы при наличии более ранних данных в область публикуемых данных передается актуальная версия данных с перемещением предыдущей версии в каталог хранения истории с партиционированием по дате загрузки.

[0026] В другом частном варианте реализации системы hCatalog предоставляет отображение структуры данных области публикации данных.

[0027] В другом частном варианте реализации системы структура данных разбивается по базам данных, соответствующих источникам предоставления данных.

[0028] В другом частном варианте реализации системы подсистема обработки и архивирования данных дополнительно обеспечивает автоматизированную функцию оката загрузки данных на любую дату в прошлом.

[0029] Заявленное решение осуществляется также с помощью способа управления большими данными (Big Data) с помощью подсистемы транспортировки и проверки входных данных (далее TFS) и подсистемы загрузки внешних данных (далее xStream), причем xStream состоит из подсистемы загрузки файлов, подсистемы обработки и архивирования, подсистемы доступа к данным, модуля управления загрузкой данных и модуля управления доступом, причем способ включает этапы, на которых:

с помощью модуля управления загрузкой xStream выполняют взаимодействие с TFS для получения данных от упомянутых источников, причем источники данных предварительно регистрируются в модуле управления загрузкой данных;

получают данные из упомянутых источников с помощью TFS, которая принимает данные в архивированном виде и осуществляет накопление и проверку данных, в случае успешной проверки упомянутых данных осуществляют их передачу в подсистему загрузки данных по транспортному протоколу;

осуществляют с помощью подсистемы обработки и архивирования данных обработку получаемых данных, которая включает накопление файлов, проверку файлов, распаковку архивных файлов, прошедших проверку и преобразование распакованных файлов в формат parquet;

осуществляют структуризацию преобразованных файлов с помощью их размещения в каталогах, каждый из которых связан с источниками данных, зарегистрированными в упомянутом модуле управления загрузкой данных;

осуществляют контроль и удаление дублирующих данных, контроль и удаление данных с нарушенной структурой, преобразование типов данных в Hive-SQL, контроль обновление каталога актуальных данных, контроль и обновление каталога истории изменения данных, контроль и управление глубиной архива данных;

осуществляют доступ пользователей к данным, размещенным в подсистеме доступа к данным с помощью модуля управления доступом.

[0030] В одном из частных примеров реализации способа модуль управления доступом определяет набор функционала на основании уровня доступа пользователя.

[0031] В другом частном примере реализации способа подсистема обработки и архивирования данных осуществляет обработку файлов формата parquet для их соответствия типу Hive-SQL.

[0032] В другом частном примере реализации способа регистрация источников данных осуществляется с помощью записи ID источника в модуле управления загрузкой.

[0033] В другом частном примере реализации способа модуль управления загрузкой осуществляет управление потоком данных множества источников по соответствующим сохраненным ID.

[0034] В другом частном примере реализации способа управление потоком данных включает проверку наличия информации от источника данных в TFS, обработку сообщений от TFS, выполнение действий на основании обработки сообщений.

[0035] В другом частном примере реализации способа выполняется полная или инкрементальная загрузка данных из подсистемы первичной обработки входных данных.

[0036] В другом частном примере реализации способа инкрементальная загрузка выполняется при обнаружении модулем загрузки наличия новых данных.

[0037] В другом частном примере реализации способа для каждого источника данных в модуле управления загрузкой содержатся параметры загрузки данных.

[0038] В другом частном примере реализации способа загрузка файлов в подсистему обработки и архивирования осуществляется на основании маски загрузки файлов.

[0039] В другом частном примере реализации способа маска загрузки файлов формируется на основании, по меньшей мере, имени загруженного архивного файла.

ОПИСАНИЕ ЧЕРТЕЖЕЙ

[0040] Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания и прилагаемых чертежей, на которых:

[0041] Фиг. 1 иллюстрирует заявленную систему комплексного управления большими данными.

[0042] Фиг. 2 иллюстрирует пример идентификатора источника.

[0043] Фиг. 3 иллюстрирует общий вид вычислительного устройства пользователя.

[0044] Фиг. 4 иллюстрирует общую схему сетевого взаимодействия.

ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

[0045] На Фиг. 1 представлена общая схема реализации заявленной системы (100) управления большими данными. Основным функциональным элементом заявленной системы (100) является подсистема (170) загрузки внешних данных – xStream, которая взаимодействует (или является частью) с облачной платформой (ОП) (180) хранения и управления данными. XStream представляет собой фреймворк для обеспечения централизованной инфраструктуры получения, обработки и хранения внешних данных в ОП на Hadoop для дальнейшего распространения всем заинтересованным лицам с целью использования в бизнес-процессах и в исследованиях на предмет извлечения ценности.

[0046] Источники данных (110) могут представлять различные ресурсы и каналы предоставления информации, например, социальные сети, системы бухгалтерского учета, CRM-системы, реестры госорганов (ЕГРЮЛ, РОССТАТ, ФСФР и др.) и т.п.

[0047] Информация, поступающая из источников данных (110), первоначально обрабатывается подсистемой транспортировки и проверки входных данных (далее TFS/Transport File System) (120). TFS (120) осуществляет прием, проверку и передачу по

транспортному протоколу в xStream (170) данных, поступающих от источников данных (110). В качестве протокола передачи информации от TFS (120) в xStream (170) может применяться протокол сетевого доступа к файловым системам (NFS - Network File System). Данные поступают на вход TFS (120) в архивированном виде.

[0048] Подсистема xStream (170) в свою очередь состоит из: подсистемы загрузки файлов (130), подсистемы обработки и архивирования данных (140), подсистемы доступа к данным (150) и технологического ядра (160).

[0049] Данные, поступившие на вход TFS (120), передаются в подсистему загрузки файлов (130), которая выполняет транспортировку архивов с данными с помощью модуль интеграции (131) с TFS (120) в первичный слой хранения данных – подсистему (140).

[0050] Для каждого из источников данных (110) предполагается два типа загрузки:

- Первичная загрузка – является инициализирующей загрузкой, которая выполняется один раз и захватывает все архивы доступные в TFS (120), названия файлов которых удовлетворяют спецификациям на поставку данных от поставщиков.
- Регулярная загрузка – выполняется инкрементально, при которой захватываются только архивы, которые не были загружены ранее.

[0051] Захват архивов происходит из каталогов ТФС. При доступе к TFS (120) осуществляется аутентификация с помощью технической учетной записи и приватного ключа. Копирование данных из TFS (120) в первичный слой хранения xStream (170) осуществляется с помощью инициирования модулем управления загрузкой (161) запросов данных из TFS (120). В TFS (120) указываются маршруты загрузки данных и управляющих сообщений между модулем (161) и TFS (120). Модуль (161) может инициировать загрузку одного или нескольких поток данных одновременно, при этом потоки будут обрабатываться параллельно.

[0052] Каждый источник данных (110) регистрируется в модуле (161). Для каждого источника создается и хранится его идентификатор (ID). В процессе регистрации источника данных (110) выполняется следующий порядок действий:

- Присвоение номера источнику данных (110);
- Определение схем внутри источника (110) и присвоения им номера. Схема является необязательным элементом и применяется для логической группировки таблиц.
- Присвоение номера всем таблицам внутри источника (110) в пределах схемы (при условии ее использования).

[0053] Источник (110), схемы, таблицы соответствуют в модуле (161) сущностям с уникальными идентификаторами. Идентификатор представляет собой целое число формата, представленного на Фиг. 2.

[0054] Номер источника – это внутренний суррогатный идентификатор источника (110) в xStream (170), который генерируется на этапе подключения источника (110) к xStream (170).

[0055] Область данных указывает на область хранения получаемой информации в xStream (170):

- 1 – область публикуемых данных (143);
- 2 – область сырых данных (141);
- 3 – область промежуточных данных (142);
- 0 – базовая сущность, идентифицирующая данные источника.

[0056] Схема представляет собой идентификатор схемы в источнике или подсистемы. Генерируется на этапе подключения источника (110) к xStream (170). Номер таблицы представляет номер таблицы в схеме., также генерируется на этапе подключения источника (110) к xStream (170).

[0057] Подсистема обработки и архивирования данных (140) представляет собой хранилище данных, получаемых от внешних источников (110), и является логической областью в HDFS. Доступ к хранимым данным осуществляется через подсистему доступа (150) hCatalog, которая также предназначена для публикации метаинформации о данных.

[0058] Подсистема обработки и архивирования данных (140) содержит области хранения данных, которые осуществляют структурированное хранение первичных (сырых) данных (141), промежуточных данных (142) и публикуемых данных (143).

[0059] В область хранения первичных данных (141) передаются архивы информации, поучаемые от TFS (120) из источников данных (110), зарегистрированных в модуле (161). Получаемые данные проверяются на целостность в TFS (120). В случае нарушения целостности архивов, получаемых от источников данных (110), такая информация не передается в подсистему xStream (170). При успешном копировании информации из TFS (120) в область первичных данных (141), xStream (170) уведомляет о проведении успешной операции.

[0060] При загрузке данных в область хранения промежуточных данных (142) первоначально выполняется разархивация архивов данных из области первичных данных (141). Передача файлов из подсистемы (130) в подсистему хранения (140) может осуществляться с помощью маски загрузки файлов, которая может формироваться на основании, например, имени (названия) загруженного архивного файла. В системе (100) сохраняется информация обо всех успешно загруженных архивах из источников данных (110).

[0061] Модуль ETL (Extract, Transform, Load) (132) выполняет передачу данных в необходимые области (142)-(143) подсистемы хранения и архивации (140), а также осуществляет подготовку и преобразование разархивированных данных в формат parquet при их поступлении в подсистему (130). Parquet представляет собой бинарный, колоночно-ориентированный формат хранения данных (см, например, «Производительность Apache Parquet». // https://habr.com/post/282552/).

[0062] При преобразовании данные источников (110) приводятся к типам Hive-SQL. Пример приведения данных к типу Hive-AQL представлен в Таблице 1. В случае если наименование файла, например, формата *.json, не соответствует маске, то данные архивов, содержащих такие файлы, не будут преобразованы в parquet. Данная методика обеспечивает устойчивый процесс обращения к данным через Hive с помощью обычных SQL-запросов, что приводит к повышению надежности доступа к информации.

Таблица 1. Матрица конвертации типов данных

Типы данных, тех. поля	**Типы данных Hive-SQL**
D(10)	STRING
F	BOOLEAN
N(1)	BIGINT
N(5,2)	DECIMAL(22,2)
N(6)	BIGINT
N(8)	BIGINT
N(9)	BIGINT
N(13)	BIGINT
N(15)	BIGINT
N(18,2)	DECIMAL(22,2)
N(19)	DECIMAL (22,0)
N(22,2)	DECIMAL (22,2)
N(22)	DECIMAL (22,0)
N(1000)	DECIMAL (38,0)
T(1)	STRING
T(3)	STRING
T(4)	STRING
T(5)	STRING
T(6)	STRING
T(7)	STRING
T(8)	STRING
T(9)	STRING
T(10)	STRING
T(11)	STRING

Типы данных, тех. поля	Типы данных Hive-SQL
T(12)	STRING
T(13)	STRING
T(15)	STRING

[0063] В область публикуемых данных (143) передаются подготовленные, преобразованные, унифицированные данные, которые приводятся к стандарту подсистемы xStream (170). При подготовке унифицированных данных имена таблиц и полей приведены к стандарту xStream, типы данных приведены к стандарту xStream, таблицы дополнены обязательными техническими полями, описывающими поставку данных - датами начала и конца периода актуальности поставки данных, номерами потоков, выполнивших загрузку и преобразование

[0064] Выполнение вышеуказанных процедур по обработке данных в подсистеме (140) позволяет организовать и обновлять актуальный срез данных (например, исторический срез) для каждого источника (110) в первичном слое xStream (170), при этом также обеспечивается поддержание требуемой временной глубины архива данных.

[0065] Загрузка данных в область публикации (143) осуществляется в несколько этапов. На первом этапе выполняется получение новых архивов. Для получения нового списка архивов сканируется область хранения сырых данных (141) по имени соответствующего источника (110) (например, для источника abc – каталог в области (141) /data/core/external/abc/src) по шаблону имени архивов. Из полученного списка исключаются архивы, которые уже были загруженные в область публикации (143) источника, которые были успешно обработаны подсистемой хранения и архивирования (140). Также, из полученного списка исключаются архивы, которые были обработаны с ошибкой. Архивы источника (110) из оставшегося списка переносятся в область сырых данных (141).

[0066] Далее осуществляется распаковка полученных архивов. Для распаковки, выбираются только те архивы из области (141), наименование которых соответствует шаблону. Из полученного списка исключаются архивы источника (110), которые ранее были загружены в область публикации (143) данного источника. Из полученного списка исключаются архивы, которые были обработаны с ошибкой. Архивы из оставшегося списка распаковываются в область промежуточных данных (142) в соответствующий подкаталог, например, src/<loading_id>/unpack/<имя_архива>.

[0067] Каждый архив распаковывается в отдельный подкаталог в области промежуточных данных (142), соответствующий названию архива. В случае успешной распаковки архива, его имя регистрируется в файле src/<loading_id>/unpack/.success, который представляет собой журнал успешной загрузки,

при возникновении ошибки при распаковке архива, его имя зарегистрируется в файле src/<loading_id>/unpack/.fail, который является, соответственно, журналом неуспешной загрузки. Обработка других архивов при получении данных при этом не прерывается.

[0068] После сохранения данных из архивов в области промежуточных данных (142) выполняется их преобразование в формат parquet. Переводу в формат parquet подлежат только те файлы данных из архивов, которые были зарегистрированы в подсистеме (140) как успешно распакованные. При этом файлы из разных архивов обрабатываются отдельно. Каждый файл из архива соответствует только одной сущности, например, таблице источника, в случае если данные были загружены из табличного вида (определяется по маске имени файла), файлы одного архива, соответствующие одной таблице, обрабатываются вместе. Таблицы дополняются служебными атрибутами (см. Таблицу 2), которые позволяют поддерживать актуальный снимок (слепок) и формировать историю изменения файлов, полученных от источника данных (110).

Таблица 2. Пример служебных атрибутов

Атрибут	Тип	Описание
ctl_loading	Int	Идентификатор загрузки модуля (161), в рамках которого запись была вставлена или обновлена
ctl_pa_loading	Int	Значение атрибута ctl_loading из Реплики перед переносом записи в Историю. Значение null означает, что записи с таким состоянием в Реплике не было. ctl_pa_loading < ctl_loading
ctl_action	String	Индикатор последней операции над записью в источнике: ▪ I – запись была вставлена ▪ U – обновлена ▪
ctl_validFrom	String	Временная метка вставки/обновления. Время подтверждения (commit-а) транзакции, в рамках которой запись попала в snp.
ctl_validTo	String	Временная метка переноса записи из snp в history. Поле ctl_validTo может быть больше или равно ctl_validFrom. Равенство возможно, если от источника

Атрибут	Тип	Описание
		данные пришли за предыдущий период (задним числом), в таком случае запись сразу попадает в history.
archive_name	String	Наименование архива поставщика, в котором находится файл, содержащий данные этой таблицы (Например: abc_fb_pagesgeneralinfo_<date>_<time>_nnn.tar.gz).
file_name	String	Наименование файла, который находился в архиве поставщика, содержащий данные этой таблицы (Например: abc_fb_pagesgeneralinfo_20171031_191158_001.json).

[0069] Вышеуказанные атрибуты передаются как параметры задания и обеспечиваются функционалом модуля управления загрузкой (161). В результате обработки создается соответствующая структура каталогов. В случае успешного преобразования всех файлов из архивов в формат parquet имя архива регистрируется в журнале, что необходимо для управления процессом доступа к информации и обеспечения автоматического отката системы на более раннюю точку. При возникновении ошибки при обработке хотя бы одного файла архива, последующая обработка архива прекращается и имя архива помечается как ошибочное.

[0070] Обработанные и унифицированные данные из области публикуемых данных (143) передаются в подсистему доступа к данным (150) для обеспечения получения данных для работы конечными пользователями.

[0071] В подсистеме доступа к данным (150) формируется два раздела – актуальный срез данных (151), содержащий отпечаток текущих данных, и раздел истории данных (152), содержащий информацию об изменениях данных.

[0072] Каждая сущность, содержащая данные, источника (110) обрабатывается отдельно, при этом обрабатываются данные, соответствующие сущности, из всех архивов, которые были зарегистрированы как успешно обработанные. Предусмотрены два способа обработки данных - выбор способа зависит от наличия историчности в данных по упомянутой сущности.

[0073] В случае отсутствия выделения историчности (наличие новых данных) выполняется следующее. Для всех новых данных по таблице архива, хранящегося в области

промежуточных данных (142), обеспечивается установка атрибута с текущим значением времени старта потока передачи данных. Данные сохраняются с партиционированием в области публикуемых данных (143) stg/<loadingId>/pa/snp/<имя_таблицы>, в результате чего создается только одна партиция (раздел).

[0074] В случае наличия выделения историчности (наличие новых данных) для всех новых данных по таблице из области промежуточных данных (142) устанавливается атрибут со временем старта потока загрузки данных модулем (161). Полученные данные объединяются с данными таблицы из области публикуемых данных (143) и из полученного объединения выделяются новые данные, которые помещаются в области (151) и (152).

[0075] Данные для отображения в области (151) сохраняются с партиционированием по полю ctl_loading в каталоге промежуточных данных (142) stg/<loadingId>/pa/snp/<имя_таблицы>. Ctl_loading – это поток (техническая сущность), который активируется модулем управления загрузкой (161). Под каждый источник (110) создается и регистрируется отдельный поток.

[0076] Для данных в области (152) истории изменения данных обеспечивается заполнение полей ctl_pa_loading (из поля ctl_loading), ctl_loading (текущим значением <loadingId>) и ctl_validTo (временем старта потока загрузки данных). Данные для отображения в области (152) сохраняются с партиционированием по полю ctl_loading в каталоге промежуточных данных (142) stg/<loadingId>/pa/hist/<имя_таблицы>. В результате для области (152) создается только одна партиция (раздел) stg/<loadingId>/pa/hist/<имя_таблицы>/ctl_loading=<loadingId>.

[0077] По окончании данного действия в каталог области хранения промежуточных данных (142) stg/<loadingId>/pa/ копируется файл stg/status/.fail. По окончанию операции область хранения промежуточных данных (142) очищается. Данные журнала операции, производимой в области (142), должны быть добавлены в общий журнал области публикуемых данных (143).

[0078] Далее выполняется объединение файлов из области (142) stg/status/.success и области (141) src/<loading_id>/parquet/.success в файл stg/<loadingId>/pa/.success. Этап считается успешным, если все данные при этом были успешно обработаны, иначе поток завершится с ошибкой.

[0079] В рамках реализации заявленного решения предусмотрены два способа публикации данных, полученные из источника (110), выбор способа зависит от наличия историчности изменений в данных в данном источнике (110).

[0080] Если для источника (110) отсутствуют сведения об изменениях, то каталог stg/<loadingId>/pa/snp/<имя_таблицы>/ctl_loading=<loadingId> из области хранения промежуточных данных (142) перемещается в каталог области публикуемых данных (143) pa/snp/<имя_таблицы> как новый раздел, который впоследствии регистрируется для отображения в hCatalog (150) в области (151).

[0081] В случае наличия выделения историчности (наличие новых данных) для источника данных (110) выполняется следующее. Каталог с данными для источника (110) pa/snp/<имя_таблицы> из области (143) перемещается в stg/<loadingId>/reserve/pa/snp области промежуточных данных (142). Каталог stg/<loadingId>/pa/snp/<имя_таблицы> из области промежуточных данных (142) перемещается в каталог области публикуемых данных (142) как pa/snp/<имя_таблицы>. Новый раздел в области (143) pa/snp/<имя_таблицы>/ctl_loading=<loadingId> регистрируется в hCatalog (150) для предоставления доступа к информации.

[0082] Каталог в области (142) stg/<loadingId>/pa/hist/<имя_таблицы>/ctl_loading=<loadingId>, созданный ранее, перемещается в каталог в области (143) pa/hist/<имя_таблицы> как новый раздел, который также регистрируется в подсистеме (150). По окончании данного процесса в каталог области промежуточных данных (142) stg/status/ копируются файлы stg/<loadingId>/pa/.success и stg/<loadingId>/pa/.fail, которые отображают статус выполнения операций загрузки данных. Данный этап считается успешным, если во время его исполнения не возникало исключений и ошибок загрузки данных.

[0083] Подсистема обработки и архивирования данных (140) очищает/архивирует область публикуемых данных (143) по параметру идентификатора потока загрузки информации (ctl_loading и ctl_pa_loading), который устанавливается модулем управления загрузкой (161). Глубина истории подлежащей очистке/архивированию задается параметром, например, 5 лет.

[0084] Для подключения нового источника данных (110) или изменения параметров существующего подсистема xStream (170) имеет конфигурационные файлы, а также профили, указываемые при запуске подсистемы XStream (170), содержащие набор параметров. Конфигурационные файлы предполагают настройку следующих параметров:

- адрес папки источника
- тип входных файлов,
- имена архивов и файлов в архиве,
- тип преобразований,
- необходимость ведения истории

- ролевая модель
- скрипты создания базы данных и таблиц для источника
- и др.

[0085] Файлы от отправителя – источника данных (110) как правило публикуются в общую папку отправителя в TFS (120), поэтому для этого необходимо осуществлять формирование уникального имени файла. Файлы поставляются упакованные в архивах. Допустимые типы расширения могут быть различными, например, tar.gz, zip и т.п.

[0086] Имя файла архива может формироваться, например, по следующей маске: <saller>_<source>_<table>_<inc>_<ver>_<date>_<time>_<nnn>.<extension>, где

- saller – конечный поставщик данных (или прокси-поставщик);
- source – источник данных (110);
- table – название типа данных или сущности. В случае если в поставке одним архивом приходит схема, состоящая из нескольких взаимосвязанных сущностей, которые необходимо поставлять одновременно, например, таблиц, то в этой секции ставится – kit, а в самих файлах внутри архива уже указываются непосредственные имена сущностей;
- inc – содержит full если архив относится к поставке/перепоставке полного архива данных, incr - если архив относится к поставке инкремента;
- ver – версия поставки. В случае изменения состава данных или формата поставки необходимо инкрементировать версию вверх. Может состоять из литеры v и 3-х цифр, начиная с v001, например, v001, v002 и т.д.;
- date – дата формирования файла в формате ГГГГММДД;
- time – время формирования файла в формате ЧЧММСС;
- nnn – номер по порядку, начинающийся с 001, если несколько файлов поставки в рамках совпадений остальных частей названия файла архива. Всегда состоит из 3-х цифр начиная с 001, например, 001, 002 и т.д. Если в поставке только 1 файл, то в этой секции указано – 001;
- extension – расширение архива.

[0087] Пример имени архива: abc_def_org1_incr_v002_20171106_112400_001.tar.gz. Не допускается наличие архива в архиве. Имя файла в архиве формируется также, как имя файла архива, за исключением расширения файла и секции <table> для набора.

[0088] Имя архива: abc_def_kit_incr_v002_20171106_112400_001.tar.gz, файлы в архиве: abc_def _kit1_incr_v002_20171106_112400_001.xml, abc_def

_org1_incr_v002_20171106_112400_001.xml и т.д. Допустимые типы и расширения файлов, например, csv, tsv, txt, json, avro, xml и др. В имени передаваемого файла допускается использование не более 128 символов (включая расширение).

[0089] Модель данных внешнего источника (110) соответствует структуре поставляемых данных и определяется на этапе анализа и подготовки источника (110) к загрузке в xStream (170). Целевым средством для доступа к данным служит Apache Hive.

[0090] Доступ пользователей к опубликованным данным, содержащимся в области (150) осуществляется на основании контроля уровня доступа каждого из пользователей. Для каждого пользователя при взаимодействии с xStream (170) проверяется разрешенный функционал по осуществлению операций с данными, в частности, такими операциями могут быть: просмотр, редактирование, получение аналитического среза, комбинированный просмотр и т.п. Пользователь с ролями Администратор и Аудитор имеют доступ к логам xStream (170) через централизованную систему управления логами, доступную в облачной платформе (180).

[0091] На Фиг. 3 представлен общий вид вычислительного устройства (200), используя которое в составе кластера реализуется заявленный способ и система.

[0092] В общем случае, вычислительное устройство (200) содержит объединенные общей шиной один или несколько процессоров (201), средства памяти, такие как ОЗУ (202) и ПЗУ (203), интерфейсы ввода/вывода (204), средства ввода/вывода (205), и средство для сетевого взаимодействия (206).

[0093] Процессор (201) (или несколько процессоров, многоядерный процессор) могут выбираться из ассортимента устройств, широко применяемых в текущее время, например, компаний Intel™, AMD™, Apple™, Samsung Exynos™, MediaTEK™, Qualcomm Snapdragon™ и т.п.

[0094] ОЗУ (202) представляет собой оперативную память и предназначено для хранения исполняемых процессором (201) машиночитаемых инструкций, для выполнения необходимых операций по логической обработке данных. ОЗУ (202), как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.).

[0095] ПЗУ (203) представляет собой одно или более устройств постоянного хранения данных, например, жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

[0096] Для организации работы компонентов устройства (200) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (204).

Выбор соответствующих интерфейсов зависит от конкретного исполнения вычислительного устройства, которые могут представлять собой, не ограничиваясь: PCI, AGP, PS/2, IrDa, FireWire, LPT, COM, SATA, IDE, Lightning, USB (2.0, 3.0, 3.1, micro, mini, type C), TRS/Audio jack (2.5, 3.5, 6.35), HDMI, DVI, VGA, Display Port, RJ45, RS232 и т.п.

[0097] Для обеспечения взаимодействия пользователя с вычислительным устройством (200) применяются различные средства (205) В/В информации, например, клавиатура, дисплей (монитор), сенсорный дисплей, тач-пад, джойстик, манипулятор мышь, световое перо, стилус, сенсорная панель, трекбол, динамики, микрофон, средства дополненной реальности, оптические сенсоры, планшет, световые индикаторы, проектор, камера, средства биометрической идентификации (сканер сетчатки глаза, сканер отпечатков пальцев, модуль распознавания голоса) и т.п.

[0098] Средство сетевого взаимодействия (206) обеспечивает передачу данных устройством (200) посредством внутренней или внешней вычислительной сети, например, Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств (206) может использоваться, но не ограничиваться: Ethernet карта, GSM модем, GPRS модем, LTE модем, 5G модем, модуль спутниковой связи, NFC модуль, Bluetooth и/или BLE модуль, Wi-Fi модуль и др.

[0099] Дополнительно могут применяться также средства спутниковой навигации, например, GPS, ГЛОНАСС, BeiDou, Galileo.

[0100] На Фиг. 4 представлен пример сетевого окружения при осуществлении работы заявленной системы (100). Организация работы с данными при использовании HDFS заключается в формировании соответствующих уровней абстракции в кластерных или виртуальных средах. Каждый стек системы включает множество вычислительных устройств, например, компьютеров и/или серверов, которые обмениваются данными с облачной платформой (180), содержащей xstream (170), посредством коммутаторов. Такая архитектура позволяет оперативно наращивать необходимые вычислительные мощности при значительном росте объема хранимых и обрабатываемых данных.

[0101] Представленные материалы заявки раскрывают предпочтительные примеры реализации технического решения и не должны трактоваться как ограничивающие иные, частные примеры его воплощения, не выходящие за пределы испрашиваемой правовой охраны, которые являются очевидными для специалистов соответствующей области техники.

ФОРМУЛА

1. Система комплексного управления большими данными (Big Data), содержащая подсистему транспортировки и проверки входных данных (далее TFS), подсистему загрузки внешних данных (далее xStream), функционирующую на основании стандарта описания данных, в которой:

- TFS осуществляет прием, проверку и передачу в xStream данных, поступающих от источников данных, причем TFS принимает данные в архивированном виде и осуществляет передачу данных по транспортному протоколу;
- xStream содержит подсистему загрузки файлов, подсистему обработки и архивирования данных, подсистему доступа к данным (далее hCatalog), модуль управления загрузкой данных и модуль управления доступом, причем в xStream:
    - подсистема загрузки файлов осуществляет опрос TFS для передачи данных, поступивших от источников, в подсистему обработки и архивирования данных, причем источники данных предварительно регистрируются в модуле управления загрузкой;
    - подсистема загрузки данных выполняет первичное копирование файлов, поступающих от TFS, в подсистему обработки и архивирования;
    - подсистема обработки и архивирования содержит области хранения данных, которые осуществляют структурированное хранение первичных данных, промежуточных данных и публикуемых данных;
        - в область хранения первичных данных передаются файлы из подсистемы загрузки данных, которые соответствуют установленным параметрам архивирования и хранятся в первоначально заархивированном виде;
        - подсистема обработки и архивирования выполнена с возможностью передачи в область промежуточных данных разархивированных первичных данных, преобразованных в формат parquet для дальнейших преобразований;
        - в область публикуемых данных передаются подготовленные, преобразованные, унифицированные данные, приведенные к стандарту xStream, в которых имена таблиц и полей приведены к стандарту xStream, типы данных приведены к стандарту xStream, таблицы дополнены обязательными техническими полями, описывающими поставку данных - датами начала и конца периода

актуальности поставки данных, номерами потоков, выполнивших загрузку и преобразование;

- и из унифицированных данных области публикуемых данных формируется структурированный каталог для доступа к упомянутым данным посредством hCatalog.

2. Система по п.1, характеризующаяся тем, что TFS осуществляет проверку целостности получаемых архивных данных.

3. Система по п.1, характеризующаяся тем, что для зарегистрированных источников в модуле управления загрузкой хранится ID упомянутых источников.

4. Система по п.3, характеризующаяся тем, что модуль управления загрузкой осуществляет управление потоком данных множества источников по соответствующим сохраненным ID.

5. Система по п.4, характеризующаяся тем, что для каждого источника данных в модуле управления загрузкой содержатся параметры загрузки данных.

6. Система по п.1, характеризующаяся тем, что подсистема загрузки данных осуществляет загрузку файлов в подсистему обработки и архивирования на основании маски загрузки файлов.

7. Система по п.6, характеризующаяся тем, что маска загрузки файлов формируется на основании, по меньшей мере, имени загруженного архивного файла.

8. Система по п.1, характеризующаяся тем, что в подсистеме обработки и архивирования в каждой из областей хранения данных формируется каталог для хранения данных соответствующего источника.

9. Система по п.7, характеризующаяся тем, что модуль управления загрузкой проверяет наличие информации по всем источникам в TFS.

10. Система по п.9, характеризующаяся тем, что выполняется полная или инкрементальная загрузка данных из TFS.

11. Система по п.10, характеризующаяся тем, что инкрементальная загрузка выполняется модулем загрузки данных при обнаружении в TFS новых данных отличающихся от поставленных ранее по дате поставки архива файлов.

12. Система по п.1, характеризующаяся тем, что подсистема обработки и архивирования осуществляет обработку parquet файлов для их приведения в соответствие типам Hive-SQL.

13. Система по п.1, характеризующаяся тем, что для файлов при их обработке подсистемой обработки и архивирования выполняется проверка на наличие аналогичных данных, сохраненных ранее.

14. Система по п.13, характеризующаяся тем, что при наличии более ранних данных в область публикуемых данных передается актуальная версия данных с перемещением предыдущей версии в каталог хранения истории с партиционированием по дате загрузки.

15. Система по п.1, характеризующаяся тем, что hCatalog предоставляет отображение структуры данных области публикации данных.

16. Система по п.15, характеризующаяся тем, что структура данных разбивается по базам данных, соответствующих источникам предоставления данных.

17. Система по п.1, характеризующаяся тем, что подсистема обработки и архивирования данных дополнительно обеспечивает автоматизированную функцию оката загрузки данных на любую дату в прошлом.

18. Способ управления большими данными (Big Data) с помощью подсистемы транспортировки и проверки входных данных (далее TFS) и подсистемы загрузки внешних данных (далее xStream), причем xStream состоит из подсистемы загрузки файлов, подсистемы обработки и архивирования, подсистемы доступа к данным, модуля управления загрузкой данных и модуля управления доступом, причем способ включает этапы, на которых:

- с помощью модуля управления загрузкой xStream выполняют взаимодействие с TFS для получения данных от упомянутых источников, причем источники данных предварительно регистрируются в модуле управления загрузкой данных;
- получают данные из упомянутых источников с помощью TFS, которая принимает данные в архивированном виде и осуществляет накопление и проверку данных, в случае успешной проверки упомянутых данных осуществляют их передачу в подсистему загрузки данных по транспортному протоколу;
- осуществляют с помощью подсистемы обработки и архивирования данных обработку получаемых данных, которая включает накопление файлов, проверку файлов, распаковку архивных файлов, прошедших проверку и преобразование распакованных файлов в формат parquet;
- осуществляют структуризацию преобразованных файлов с помощью их размещения в каталогах, каждый из которых связан с источниками данных, зарегистрированными в упомянутом модуле управления загрузкой данных;
- осуществляют контроль и удаление дублирующих данных, контроль и удаление данных с нарушенной структурой, преобразование типов данных в Hive-SQL, контроль обновление каталога актуальных данных, контроль и

обновление каталога истории изменения данных, контроль и управление глубиной архива данных;

- осуществляют доступ пользователей к данным, размещенным в подсистеме доступа к данным с помощью модуля управления доступом.

19. Способ по п.18, характеризующийся тем, что модуль управления доступом определяет набор функционала на основании уровня доступа пользователя.

20. Способ по п.18, характеризующийся тем, что подсистема обработки и архивирования данных осуществляет обработку файлов формата parquet для их соответствия типу Hive-SQL.

21. Способ по п.18, характеризующийся тем, что регистрация источников данных осуществляется с помощью записи ID источника в модуле управления загрузкой.

22. Способ по п.18, характеризующийся тем, что модуль управления загрузкой осуществляет управление потоком данных множества источников по соответствующим сохраненным ID.

23. Способ по п.22, характеризующийся тем, что управление потоком данных включает проверку наличия информации от источника данных в TFS, обработку сообщений от TFS, выполнение действий на основании обработки сообщений.

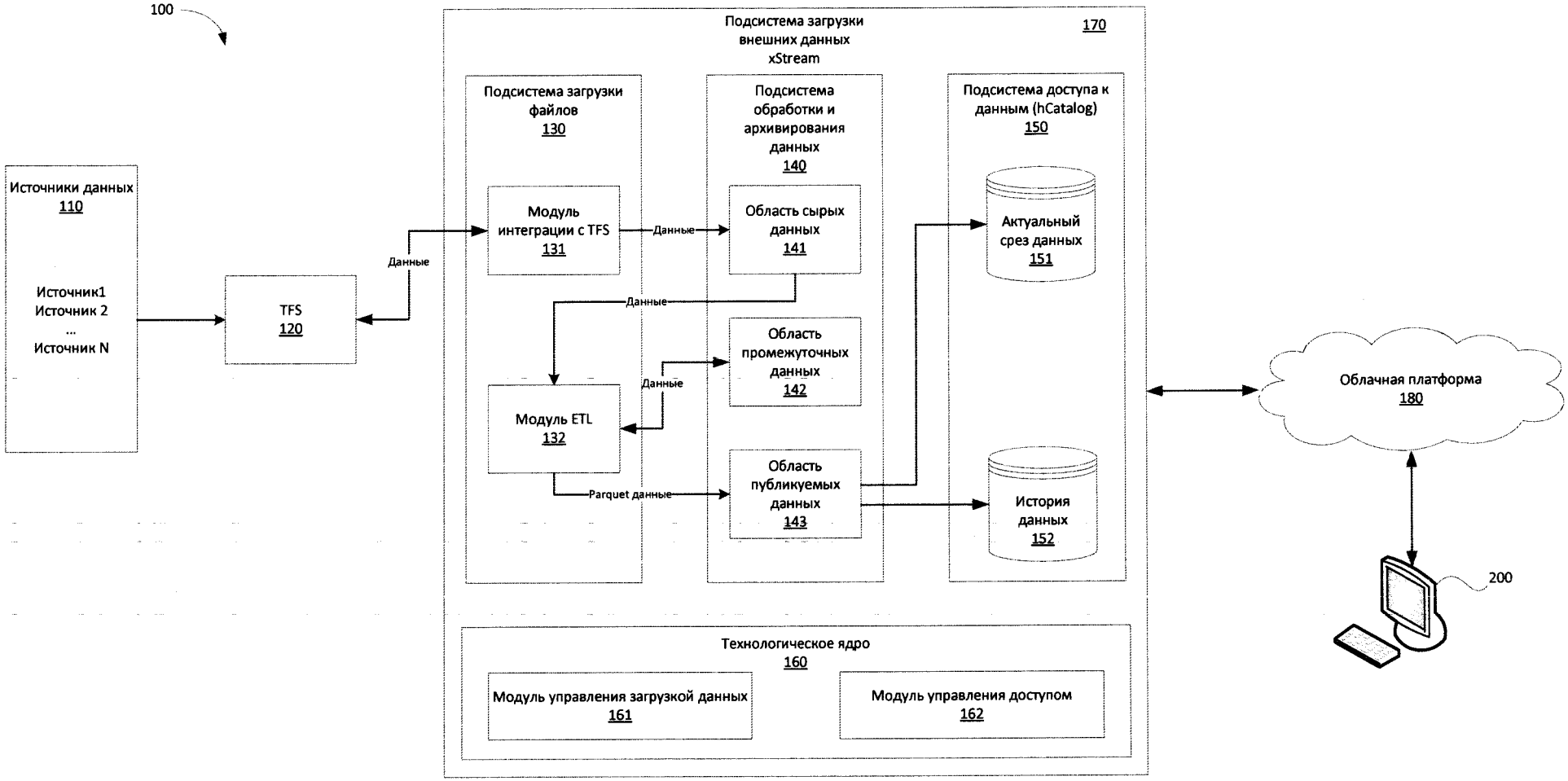
24. Способ по п.22, характеризующийся тем, что выполняется полная или инкрементальная загрузка данных из подсистемы первичной обработки входных данных.

25. Способ по п.23, характеризующийся тем, что инкрементальная загрузка выполняется при обнаружении модулем загрузки наличия новых данных.

26. Способ по п.18, характеризующийся тем, что для каждого источника данных в модуле управления загрузкой содержатся параметры загрузки данных.

27. Способ по п.18, характеризующийся тем, что загрузка файлов в подсистему обработки и архивирования осуществляется на основании маски загрузки файлов.

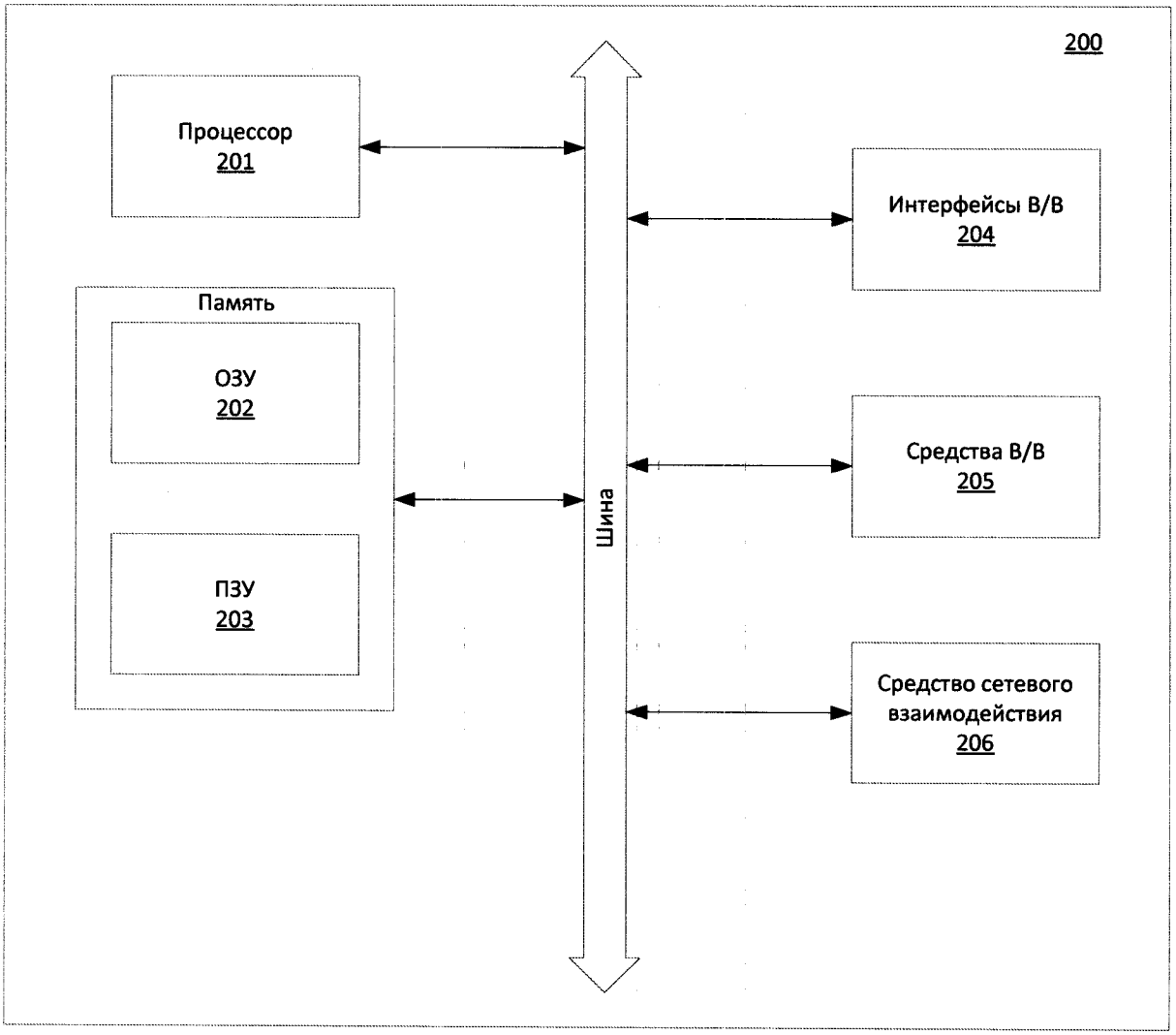
28. Способ по п.27, характеризующийся тем, что маска загрузки файлов формируется на основании, по меньшей мере, имени загруженного архивного файла.


100
Подсистема загрузки внешних данных xStream
170
Подсистема загрузки файлов
130
Подсистема обработки и архивирования данных
140
Подсистема доступа к данным (hCatalog)
150
Источники данных
110
Источник1
Источник 2
...
Источник N
TFS
120
Данные
Модуль интеграции с TFS
131
Данные
Область сырых данных
141
Актуальный срез данных
151
Данные
Область промежуточных данных
142
Данные
Модуль ETL
132
Область публикуемых данных
143
Parquet данные
История данных
152
Облачная платформа
180
200
Технологическое ядро
160
Модуль управления загрузкой данных
161
Модуль управления доступом
162

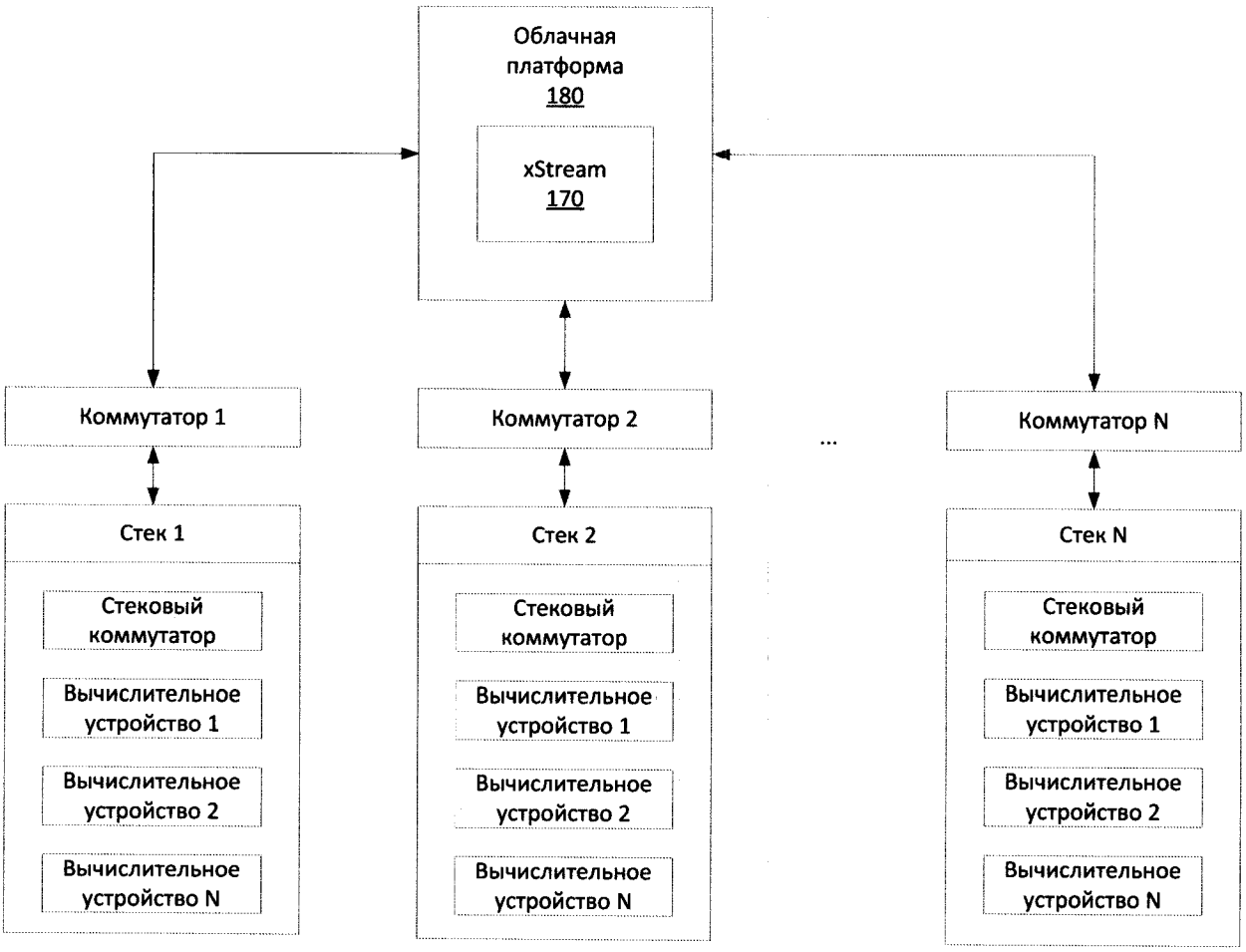
Фиг. 1


XXX
X
X
XXXX
Номер источника
Область данных
Схема
Номер таблицы

Фиг. 2


200
Процессор
201
Память
ОЗУ
202
ПЗУ
203
Шина
Интерфейсы В/В
204
Средства В/В
205
Средство сетевого взаимодействия
206

Фиг. 3


Облачная платформа 180
xStream 170
Коммутатор 1
Коммутатор 2
...
Коммутатор N
Стек 1
Стек 2
Стек N
Стековый коммутатор
Вычислительное устройство 1
Вычислительное устройство 2
Вычислительное устройство N
Стековый коммутатор
Вычислительное устройство 1
Вычислительное устройство 2
Вычислительное устройство N
Стековый коммутатор
Вычислительное устройство 1
Вычислительное устройство 2
Вычислительное устройство N

Фиг. 4

ЕВРАЗИЙСКОЕ ПАТЕНТНОЕ ВЕДОМСТВО

ОТЧЕТ О ПАТЕНТНОМ ПОИСКЕ

(статья 15(3) ЕАПК и правило 42 Патентной инструкции к ЕАПК)

Номер евразийской заявки: 201892256

Дата подачи: 02 ноября 2018 (02.11.2018) | Дата испрашиваемого приоритета: 26 октября 2018 (26.10.2018)

Название изобретения: СПОСОБ И СИСТЕМА КОМПЛЕКСНОГО УПРАВЛЕНИЯ БОЛЬШИМИ ДАННЫМИ

Заявитель: ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК)

☐ **Некоторые пункты формулы не подлежат поиску (см. раздел I дополнительного листа)**

☐ **Единство изобретения не соблюдено (см. раздел II дополнительного листа)**

А. КЛАССИФИКАЦИЯ ПРЕДМЕТА ИЗОБРЕТЕНИЯ:

МПК:	***G06F 16/11***	*(2019.01)*	СПК:	***G06F 16/11***	*(2019-01)*
	G06F 12/08	*(2016.01)*		***G06F 12/08***	*(2013-01)*
				G06F 17/00	*(2019-01)*

Согласно Международной патентной классификации (МПК) или национальной классификации и МПК

Б. ОБЛАСТЬ ПОИСКА:

Минимум просмотренной документации (система классификации и индексы МПК)
G06F 3/00, 3/06, 11/00, 12/00, 12/02, 12/08, 16/00, 16/10, 16/11, 17/00

Другая проверенная документация в той мере, в какой она включена в область поиска:

В. ДОКУМЕНТЫ, СЧИТАЮЩИЕСЯ РЕЛЕВАНТНЫМИ

Категория*	Ссылки на документы с указанием, где это возможно, релевантных частей	Относится к пункту №
A	RU 141446 U1 (САНКТ-ПЕТЕРБУРГ, ОТ ИМЕНИ КОТОРОГО ВЫСТУПАЕТ КОМИТЕТ ПО ИНФОРМАЦИИ И СВЯЗИ и др.) 10.06.2014	1-28
A	US 2016/0210064 A1 (COMMVAULT SYSTEMS, INC.) 21.07.2016	1-28
A	US 2007/0226535 A1 (PARAG GOKHALE) 27.09.2007	1-28
A	US 2013/0024424 A1 (COMMVAULT SYSTEMS, INC.) 24.01.2013	1-28
A	US 2013/0238572 A1 (COMMVAULT SYSTEMS, INC.) 12.09.2013	1-28

☐ последующие документы указаны в продолжении графы В ☐ данные о патентах-аналогах указаны в приложении

* Особые категории ссылочных документов:
"A" документ, определяющий общий уровень техники
"E" более ранний документ, но опубликованный на дату подачи евразийской заявки или после нее
"O" документ, относящийся к устному раскрытию, экспонированию и т.д.
"P" документ, опубликованный до даты подачи евразийской заявки, но после даты испрашиваемого приоритета
"D" документ, приведенный в евразийской заявке
"T" более поздний документ, опубликованный после даты приоритета и приведенный для понимания изобретения
"X" документ, имеющий наиболее близкое отношение к предмету поиска, порочащий новизну или изобретательский уровень, взятый в отдельности
"Y" документ, имеющий наиболее близкое отношение к предмету поиска, порочащий изобретательский уровень в сочетании с другими документами той же категории
"&" документ, являющийся патентом-аналогом
"L" документ, приведенный в других целях

Дата действительного завершения патентного поиска: 24 мая 2019 (24.05.2019)

Наименование и адрес Международного поискового органа:
Федеральный институт промышленной собственности
РФ, 125993,Москва, Г-59, ГСП-3, Бережковская наб., д. 30-1.Факс: (499) 243-3337, телетайп: 114818 ПОДАЧА

Уполномоченное лицо :
Т. Ф. Владимирова
Телефон № (499) 240-25-91