DOI https://doi.org/10.15407/usim.2019.02.040
УДК 303.721;004.03142

А.А. УРСАТЬЕВ, канд. техн. наук, старший научный сотрудник, ведущий научный сотрудник, Международный научно-учебный центр информационных технологий и систем НАН и МОН Украины, просп. Академика Глушкова, 40, Киев 03187, Украина, aleksei@irtc.org.ua

БОЛЬШИЕ ДАННЫЕ. АНАЛИТИЧЕСКИЕ БАЗЫ ДАННЫХ И ХРАНИЛИЩА: GREENPLUM

Статья представляет собой продолжение исследований Больших Данных и инструментария, трансформируемого в новое поколение технологий и архитектур платформ баз данных и хранилищ для интеллектуального вывода. Рассмотрен ряд прогрессивных разработок известных в мире ИТ-компаний, в частности Greenplum DB.

Ключевые слова: *Greenplum Data Computing Appliance, MPP-архитектура без разделения ресурсов на основе ядра PostgreSQL, технология MPP Scatter/Gather Streaming загрузки (выгрузки), полиморфное хранение данных, аналитика Big Data, интеграция платформ, механизм запросов SQL Hadoop HAWQ, самообслуживание данными.*

Общая характеристика Greenplum (NYSE: EMC)

Greenplum Database® — это первая в мире полнофункциональная платформа с открытым исходным кодом — проект выпущен под лицензией *Apache* 2 [90]. СУБД *Greenplum*[1] — это реляционная база данных, использующая широкомасштабную параллельную *MPP*-архитектуру без разделения ресурсов (*Shared Nothing*), на основе ядра *Postgres*[2] *Core*. Обеспечивает автоматическое распараллеливание данных и запросов в масштабируемой до размеров петабайт среде обработки. Доступ и запрос данных осуществляется через внешний синтаксис таблицы. Предназначена для управления крупномасштабными аналитическими хранилищами данных и рабочими нагрузками бизнес-аналитики и науки о данных.

Масштабируемая платформа реляционных БД *Greenplum Database* разработана компанией *Greenplum Software* (основана в 2003 г.), впоследствии (июль 2010 г.) приобретенной корпорацией *EMC*. Торговая марка «*Greenplum*» для СУБД была сохранена, а разработка передана в 2012 г. подразделению *Pivotal* — независимой компании программного обеспечения, принадлежащей *EMC*, *VMware* и *General Electric*. *EMC* вышла на рынок хранилищ данных и аналитики, представив *Greenplum Database* 4.0 и *EMC Greenplum Data Computing Appliance* (*DCA*), и позже — аналитическую БД *Pivotal Greenplum Database* корпоративного класса с мощной и быстрой аналитикой для больших объемов данных под торговой маркой *Pivotal*. Рыночный потенциал и необходимость иметь эффективную технологию для работы в пространстве хранилища данных, вероятно, явились ключе-

[1] — http://dbdb.io/db/greenplum

[2] – объектно-реляционная база данных с более чем 30-летней разработкой, заслужившая высокую репутацию за надежность, функциональность и производительность - https://www.postgresql.org/

выми факторами, которые привели к тому, что *EMC* приобрела *Greenplum Software* [30, 91–98].

Автор уже обращался к теме специализированных хранилищ и устройств обработки данных, представляющих собой сконфигурированный, подготовленный к быстрому использованию, автономный аппаратно-программный комплекс (*Appliances*), объединяющий хранение и обработку данных в одной системе, спроектированной и оптимизированной под принятие аналитических решений [98, 99]. Обусловлено это было (конец 90-х годов XIX ст. — начало 2000 годов) возрастающими требованиями информационной поддержки принятия решений на фоне стремительно увеличивающегося роста Больших Данных (*Big Date*) [30, 100].

Во-первых, широко распространенные критически важные системы оперативной обработки транзакций (*OLTP*), управляемые реляционными базами данных (*RDBMS*), были ориентированы на транзакционную обработку и не были сосредоточены на процессах принятия управленческих решений. *RDBMS* — оперативная обработка транзакций и хранение информации, где необходим быстрый доступ и обновление одиночных записей, оказались не эффективными при выполнении рабочих нагрузок бизнес-аналитики на больших наборах данных. Традиционные поставщики *RDBMS* переработали программное обеспечение баз данных *OLTP*-систем для поддержки рабочих нагрузок в *OLAP*-стиле. Хотя эти решения достаточно хороши для *OLTP*, их очень сложно создать, управлять и настраивать с учетом будущего роста хранилищ данных [92, 99].

Во-вторых, скорость загрузки данных в систему *Business Intelligence/Data Warehousing* (*BI/DW*), иначе пропускная способность хранилища, не должна препятствовать бесперебойной работе инструментов *BI*. Скорость, с которой данные могут быть загружены в систему *DW/BI*, важна для пользователей, которые стремятся перейти к анализу в режиме реального или близкого к нему времени, при существующей пакетной передаче данных в хранилище с небольшим окном загрузки. Поэтому, в зависимости от существующих операционных задач, большинство клиентов выбирают только одну область производительности: загрузка данных, запрос или работа (обработка транзакций).

Традиционные хранилища данных являются относительно медленным производителем информации для бизнес-пользователей, которым нужны полные аналитические решения для выполнения своей работы. Хранилища обычно обновляются периодически, чаще всего еженощно. Таким образом, между бизнес-транзакцией и ее появлением в *DW* существует некоторая задержка. Свежие релевантные данные попадают в операционные хранилища данных, в то время как оно недоступно для анализа бизнес-пользователей в реальном времени. Традиционный подход к хранилищу данных часто слишком медленный для быстрого принятия решений в организациях. Более того, традиционная модель *DW* может не включать в себя всю соответствующую информацию, доступную для бизнеса.

Обусловлено это проблемами увеличения объема и типа данных, количества пользователей и сложности анализа. Именно *DW* в первую очередь становится чувствительным к времени бизнес-практики, такой как операционная бизнес-аналитика (*BI*), часто обновляемые панели управления, выработка рекомендаций по электронной коммерции *на лету* и др. Бизнес-аналитики вынуждены уменьшать свои наборы данных в случае медленной загрузки данных. В конечном итоге эта ситуация приводит к менее чем оптимальному пониманию бизнеса, поскольку в анализе отсутствуют все данные, доступные для бизнес-пользователей [92, 93, 95].

Преодолеть препятствия основных бизнес-требований *DW/BI* и обеспечить предсказуемую производительность и масштабируемость должны были решения, положенные в основу упомянутых выше разработок *EMC*. *Greenplum DCA*, как и подобные устройства других производителей, относятся к подмножеству специализированных решений, предназначенных для аналитической работы с большими массивами данных. *DCA* — это интегрированная платформа аналитики, ускоряющая анализ

активов данных в рамках единого устройства с использованием базы данных *Greenplum™* для оптимизации аналитики *SQL* на структурированных данных (представлена в окт. 2010 г.). В ее основе — БД *Greenplum*, разработанная для управления крупномасштабными хранилищами данных и аналитическими рабочими нагрузками. *Greenplum* также придерживается концепции переноса аналитической обработки значительно ближе к данным и параллельным вычислениям на архитектурных принципах *MPP* [90–94, 98].

Логическая архитектура БД *Greenplum (GPDB)* [91–93, 101] представляет собой массив экземпляров (*Instances*) распределенной БД *PostgreSQL*, работающих совместно как одна СУБД, для представления единого образа БД. *GPDB* обеспечивает хранение и обработку данных, распределяя нагрузку на серверы или хосты (рис. 17). На каждой машине запущен экземпляр базы данных *PostgreSQL*, который модифицирован для поддержки параллельного выполнения запросов. Все узлы подключены через сеть *Greenplum* (*GNet Interconnect*) для формирования единого экземпляра *GPDB*. Программное обеспечение *gNet Software Interconnect* настроено и оптимизировано для масштабирования до десятков тысяч процессоров и использует отраслевые стандарты *Gigabit Ethernet* и 10 *GigE*. Двухпортовая коммутация 10 *GigE Ethernet* обеспечивает высокоскоростное, расширяемое решение *IP*-взаимодействия в рамках *GPDB*. Каждый сервер сегмента (*Segment Host*) имеет выделенный набор хранилищ с прямым подключением для минимизации задержки ввода-вывода от сервера к хранилищу. Главная база данных (*Master Database*) выполняется на основном сервере *Master Host* (реплицируется на резервный мастер-сервер) и является точкой входа в БД *Greenplum* для конечных пользователей и инструментов *BI*. Взаимодействие с БД *Greenplum* осуществляется посредством клиентских программ, таких как ***psql*** (*pgSQL*) или через интерфейсы прикладного программирования (*API*): *JDBC* или *ODBC*. Мастер-сервер координирует работу с другими экземплярами БД в системе, называемыми сегментами (*Segment Instances*).

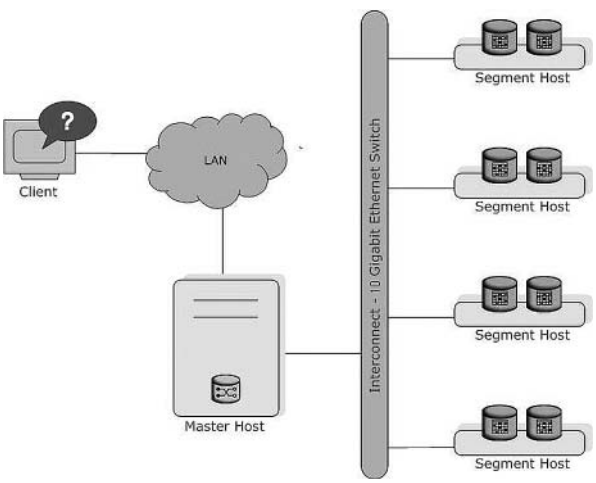

Рис. **17**

Master Host содержит глобальный системный каталог (набор системных таблиц с метаданными о данных *GPDB*), но не содержит никаких пользовательских данных. Данные хранятся только на серверах сегментов. Мастер обрабатывает входящие команды *SQL* запросов, распределяет рабочие нагрузки между экземплярами сегмента, координирует результаты, возвращаемые каждым сегментом, и предоставляет конечные результаты клиентской программе.

Segment Host. Когда запрос отправляется на *Master Host*, он оптимизирован и разбит на более мелкие компоненты, которые отправляются в сегменты для совместной работы по доставке конечных результатов. Сегментные серверы (сегменты хостов, рис. 17) запускают экземпляры базы данных — *Segment Instances*. Экземпляр сегмента — это комбинация процесса, памяти и содержимого базы данных на сервере сегментов (*Segment Servers*). На экземпляре сегмента содержится уникальный раздел всей БД. Большая часть обработки запросов происходит либо внутри сегмента, либо между экземплярами сегмента. Каждый сервер сегментов запускает несколько экземпляров сегмента — БД *Greenplum* обычно содержит от двух до восьми сегментов, в зависимости от количества ядер процессора, объема опе-

ративной памяти, хранилища, сетевых интерфейсов и рабочих нагрузок. Пользовательские таблицы и их индексы распределяются по доступным сегментам в системе *GPDB*. Каждый сегмент содержит свою часть данных. Таким образом, запросы параллельно обслуживаются в соответствующих экземплярах сегмента на каждом сервере сегментов. Пользователи взаимодействуют с сегментами *GPDB* через *Master Database*.

БД Greenplum и PostgreSQL. Основные различия

Внутренние элементы *PostgreSQL* были модифицированы или дополнены для поддержки параллельной структуры БД *Greenplum*. Так, системный каталог, планировщик запросов, оптимизатор, исполнитель запросов и компоненты диспетчера транзакций были изменены и расширены, чтобы была возможность выполнять запросы одновременно во всех параллельных экземплярах базы данных *PostgreSQL*. Сетевой уровень соединения *Greenplum* обеспечивает связь между отдельными экземплярами *PostgreSQL* и позволяет системе вести себя как одна логическая база данных. БД *Greenplum* также содержит функции, предназначенные для оптимизации рабочих нагрузок *PostgreSQL* для бизнес-аналитики (*BI*). Например, *Greenplum* добавил параллельную загрузку и полиморфное хранение данных любой таблицы или раздела, управление ресурсами, декларативные разделы и подразделы для неявного создания ограничений раздела, оптимизацию запросов и улучшения хранилища, которые не найдены в стандартном *PostgreSQL*.

Параллельная загрузка данных. В крупномасштабном хранилище традиционно данные вынуждены загружаться в относительно небольшом окне обслуживания. Компания *Greenplum* производит высокопроизводительную параллельную загрузку посредством новой технологии потоковой передачи данных *Scatter/Gather Streaming* (*SG Streaming*), используя механизм *MPP*-структуры. Подход к загрузке — *параллельно-всюду* (*parallel-everywhere*) реализуется путем *рассеивания* данных из исходных систем через параллельные потоки, одновременно поступающие ко всем узлам *GPDB*. Финальный сбор и хранение данных на дисках происходит на узлах одновременно. Технология *SG Streaming* управляет потоком данных в узлах БД, устраняя традиционные узкие места массовой загрузки, и поддерживает непрерывные модели загрузки в режиме реального времени с незначительным воздействием на одновременные операции с БД. Возможно преобразование данных (*ELT* или *ETL*) во время загрузки. Это отличает *SG Streaming* от традиционных технологий массовой загрузки, чреватой появлением *бутылочных горлышек*, из-за приема данных из одного источника, часто по одному или небольшому числу параллельных каналов, что приводит к потерям времени и постоянно увеличивающимся нагрузкам. При этом устраняется необходимость в дополнительном уровне серверов, функционирующих только как загрузчики хранилища данных. При этом они добавляют значительную сложность и стоимость, но эффективно повышают пропускную способность [93–101].

Greenplum считает, на основе отзывов клиентов, что скорость загрузки данных с использованием этой технологии в реальных производственных средах, достигает более четырех терабайт в час при незначительном воздействии на одновременные операции с базой данных [101]. БД *Greenplum* поддерживает быструю параллельную загрузку данных [91–93, 96, 102] посредством функций «*внешних таблиц*» [3]. Используя внешние таблицы (рис. 18)

[3] *Defining External Tables*. Внешние таблицы (ВТ) позволяют получать доступ к внешним данным, как к обычной таблице базы данных. ВТ представляет собой таблицу БД *Greenplum*, в которой хранятся данные, находящиеся за пределами базы данных. Часто используются для перемещения данных в БД *Greenplum* и из нее. ВТ могут быть основаны на файлах или на веб-сайтах. Внешние веб-таблицы в Интернете обеспечивают доступ к динамическим источникам данных, обслуживаемым сервером *HTTP* или операционным процессом, и позволяют БД *Greenplum* обрабатывать эти данные также как обычные таблицы БД — https://gpdb.docs.pivotal.

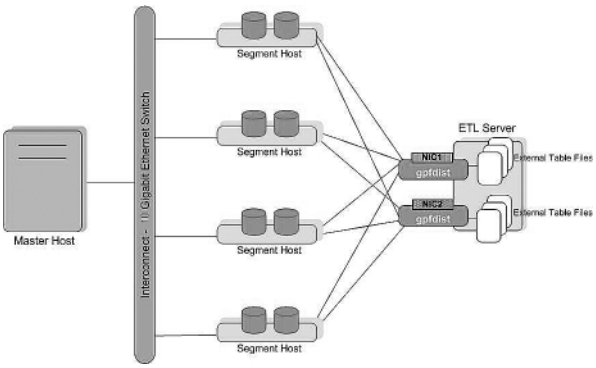

***Рис*. 18**

в сочетании с параллельным файловым сервером БД *Greenplum* (***gpfdist***), достигают максимального параллелизма и высокой пропускной способности хранилища при оптимальном использовании всех сегментов.

Полиморфное или гибридное хранение данных (*Polymorphic Data Storage™*). БД *Greenplum* может использовать оптимизированное для приложений хранилище, предоставляя гибкость выбора между хранением данных на основе колонок или строк в рамках одной и той же системы баз данных. По умолчанию в БД *Greenplum* используется модель хранения *Heap*, что и в *PostgreSQL*. Таблицы *Heap* лучше всего подходят, например, для небольших таблиц измерений, которые часто обновляются после их первоначальной загрузки. Для таблиц *Heap* доступна одна ориентация: строчная. Такая ориентация лучше всего отвечает рабочим нагрузкам типа *OLTP*, где требуется частая загрузка данных. Ориентированные на колонки таблицы не оптимизированы для операций записи, так как значения столбцов для строки должны быть записаны в разные места диска. Колоночное хранилище, например, полезно для рабочих нагрузок хранилища с совокупностью данных, рассчитанных на небольшое количество столбцов [103, 104].

Для массовой загрузки и чтения данных в БД *Greenplum* используется *Append*–оптимизированный (*AO*, *append-optimized* или *append-only*) формат хранения, обеспечивающий преимущества производительности в сравнении с таблицами *Heap*. Таблица *append-only* — это представление хранилища, которое допускает только добавление новых строк в таблицу, но исключает обновление или удаление существующих. Модель такого хранилища подходит для больших денормализованных таблиц фактов, которые загружаются с использованием пакетного процесса и доступны по запросам только для чтения. Перемещение больших таблиц фактов в модель хранения, оптимизированную для добавления, исключает издержки на сохранность информации о видимости обновлений для каждой строки, экономя около 20 байтов на строку. Это позволяет использовать более компактное хранилище на диске, поскольку не требуется сохранять информацию о транзакциях *MVCC*.

Пространство хранилища, занимаемое строками, которые обновляются и удаляются в *AO*-таблицах, не восстанавливается и не используется так же эффективно, как в *append-only* таблицах, поэтому модель *AO*-хранилища не подходит для часто обновляемых таблиц (*Heap*). Она в основном предназначена для таблиц, которые загружаются один раз, обновляются редко и часто запрашиваются для обработки аналитических запросов [92, 103, 104].

Append-оптимизированное хранилище обеспечивает контрольные суммы для защиты данных, сжатия и ориентации строк/столбцов. Обе таблицы, ориентированные на строки или столбцы, могут быть сжаты. В соответствии с принятой ориентацией данных в таблицах, разделов хранения, параметры выполнения и компрессии могут быть настроены. Возможно разделение таблиц на нескольких уровнях. Так, одну таблицу можно разделить на вертикальные разделы, часть из которых будет храниться в виде строк, а часть — как колоночные объекты. При этом для пользователя такая таблица будет выглядеть одним объектом. Ориентированное на столбцы хранилище таблиц доступно только в *AO*-таблицах.

Существует два типа компрессии данных в БД *Greenplum* для *AO*-таблиц: сжатие на уровне строк применяется ко всей таблице; сжатие на уровне колонок — к определенным столбцам. Можно применять различные доступные

алгоритмы сжатия на уровне колонок для разных столбцов в одной таблице. Эти настройки не являются общесистемными, на одной платформе можно задать несколько стратегий [103, 104].

Журнал опережающей записи WAL (*Write-Ahead Logging*) — это способ улучшения надежности БД и восстановления данных после сбоев. В БД *Greenplum* используется технология опережающей записи *WAL* для мастер-зеркалирования *Master/Standby*. В протоколе *WAL* все изменения записываются в конец файла журнала перед применением, т.е. только после их занесения в журнал, чтобы обеспечить целостность данных для любых операций внутри процесса. Все изменения записываются последовательно и, как следствие, не сильно замедляют процесс. Так, при отказе одного или нескольких сегментов они помечаются как сбойные и вместо них запускаются их зеркальные сегменты, репликация данных для которых происходит с помощью используемой технологии опережающей записи. При импорте больших файлов для аналитики, возможно отключиться *WAL* для отдельных таблиц и получить более высокую производительность. Ведение журнала *WAL* еще недоступно для зеркалирования сегментов версии БД *Greenplum* 5.8 [96].

Управление параллелизмом в БД Greenplum. Ключом к достижению наилучшей производительности есть равномерное распределение данных и рабочих нагрузок по большому числу равнозначных сегментов, чтобы они совместно работали над задачей и выполняли работу одновременно. *PostgreSQL* и БД *Greenplum* не используют блокировки для контроля параллелизма. Согласованность данных поддерживается с использованием мультиверсионной модели контроля конкурентных транзакций *MVCC* (*Multiversion Concurrency Control*), которая обеспечивает изоляцию транзакций для каждого сеанса БД. При запросах к БД каждая транзакция считывается из моментального снимка данных (версии), который не изменяется никакими параллельными транзакциями. Это гарантирует, что транзакция считает согласованные данные, на которые не влияют другие параллельные транзакции.

Поскольку *MVCC* не использует явные блокировки для управления параллелизмом, конфликт блокировки минимизируется, а БД поддерживает разумную производительность в многопользовательских средах. Вместе с тем, БД *Greenplum* предоставляет несколько режимов блокировки для управления одновременным доступом к данным в таблицах. Большинство команд автоматически получают соответствующие блокировки, чтобы гарантировать, что ссылочные таблицы не будут отброшены или изменены несовместимыми способами во время выполнения команды. Для приложений можно использовать команду *LOCK* для получения явных блокировок. Однако использование *MVCC* обычно обеспечивает лучшую производительность [105].

Разграничение зон видимости данных и прав доступа в более поздних моделях обеспечивается благодаря ролевой модели доступа (*Role Based Access Control, RBAC*), позволяющей реализовать правила, динамически меняющиеся в процессе функционирования платформы хранения и обработки данных. Так, например, можно создать схемы ограничения доступа к таблицам и другим объектам *GPDB*, а также к строкам и столбцам отдельных таблиц.

GPORCA[4] [106, 197] — оптимизатор запросов *ORCA* расширяет возможности планирования и оптимизации унаследованного *GPQUERY*. Он основан на модифицированной версии планировщика запросов БД *Postgres* и предназначен для работы с базами данных *MPP*. Стратегия оптимизации запросов, нашедшая отражение в 90-х годах XX ст. в разработках *Volcano/Cascades Optimizers* [108–110], получила свое дальнейшее развитие в исследованиях и разработках *Greenplum/Pivotal*. Побудительным мотивом этому послужил прогрессирующий рост *Big Data* и повышенный интерес к обработке сложных аналитических запросов. *Volcano* — общецелевой стоимостной оптимизатор запросов на основе затрат, созданный на правилах

[4] *GPORCA* – это оптимизатор запросов *PQO* (*Pivotal Query Optimizer*) версии с открытым исходным кодом.

эквивалентности (над) алгебрами, использует стратегию нисходящего или целевого поиска для нахождения самого *дешевого*, малозатратного плана в пространстве возможных планов. Задействует эвристику для ограничения пространства поиска выбранными планами, параллельного их выполнения и распределения ресурсов. Стоимость измеряется в дисковых вводах-выводах. Первый реализованный механизм выполнения запросов, эффективно сочетающий расширяемость и параллельность в вычислениях. Применяя стандартный интерфейс между операторами алгебр, *Volcano* позволяет легко добавлять новые операторы и правила эквивалентности, он расширяем алгоритмами и типами данных. Запросы БД *Greenplum* задействует модель процессора запросов оптимизатора, который принимает план выполнения и употребляет его для генерации дерева физических операторов, оценки таблиц через физические операторы и предоставления результатов в ответ на запрос. Генератор вырабатывает код на основе декларативных правил, логической и физической алгебры. Оптимизация проводится с использованием поиска по веткам и границам. Эффективная поисковая система *Volcano*, основана на динамическом программировании и запоминании (кэшировании эквивалентных операторов) результатов выполнения функций для предотвращения повторных вычислений, что улучшает расширяемость оптимизатора.

Основа оптимизации запросов *Volcano/Cascades* [108, 109] базируется на системе правил эквивалентности, в которой указано, что результат конкретного преобразования дерева запросов совпадает с результатом исходного дерева запросов. Ключевым аспектом этой структуры является эффективная реализация подхода, созданного на правилах преобразования.

Оптимизатор *Cascades* — объектно-ориентированная реализация оптимизатора запросов *Volcano*. *Pivotal ORCA* — реализация автономных каскадов, поддерживает многопоточный поиск.

ORCA [110] — это современный оптимизатор запросов в стиле *Volcano* (*top-down* — оптимизация сверху вниз) на основе инфраструктуры оптимизации *Cascades* [109]. Он разработан специально для решения передовой аналитики, создания планов запросов, выполняющих сложные объединения при высокой производительности на больших массивах данных. Поступающий в *ORCA* запрос анализируется и передается мастеру БД *Greenplum* то, что он считает самым быстрым планом выполнения; при этом запрос объединяется с метаданными (например, статистика, определение таблиц, схемы и др.) и информацией о кластере базы данных для создания последовательности работ. *ORCA* может взаимодействовать[5] с несколькими БД и легко поддерживать новые операторы баз данных. Способность работать за пределами системы баз данных как автономный оптимизатор допускает его использование для поддержки продуктов с различными вычислительными архитектурами с использованием одного оптимизатора. Кроме того, основные методы оптимизации являются составными, что позволяет добавлять новые операторы, трансформации, статистические/стоимостные модели и другие методы. *ORCA* развертывает высокоэффективный многоядерный планировщик, который распределяет отдельные мелкооцененные подзадачи оптимизации в нескольких ядрах для ускорения процесса оптимизации. Обеспечивает существенное улучшение в сравнении с решениями предыдущей системы и во многих случаях предлагает ускорение запросов от 5(10)*X* до 1000*X* [110, 111]. *GPORCA* — с открытым исходным кодом, построен как внешний плагин, и доступен под лицензией *Apache* вер.2.0.

Вместе с тем, *GPORCA* не является полнофункциональным в сравнении с текущим планировщиком в *GPDB*. Когда *GPORCA* встречает оператор, который не поддерживается в настоящий момент, он возвращается к унаследованному планировщику [106, 111].

[5] Взаимодействия оптимизатора и системы баз данных – это обмен метаданными. Например, оптимизатору, возможно, необходимо знать, определены ли индексы на таблице, чтобы разработать эффективный план запроса.

Новые технологии обработки

Apache Hadoop и HDFS или MapR-FS. Впервые *EMC* анонсировала свою программу развития аналитики *Big Data* в 2011 г. В частности, была представлена комплексная стратегия дистрибуции, интеграции и поддержки ПО с открытым исходным кодом *Apache Hadoop*. В результате слияния БД *Greenplum* с *Hadoop* появилась возможность в рамках одного хранилища обрабатывать как структурированные, так и неструктурированные данные [112, 113].

Судя по технической документации на оборудование, *Greenplum* давно готовилась к развитию работ по расширению типов данных в аналитических исследованиях. Так, *GPDB v*. 3.2 выпуска 2008 г. представляет ряд новых функций, улучшений производительности и стабильности, внутренних изменений в системных каталогах. Одна из них — параллельные распределенные вычисления на *MapReduce*. *Greenplum* предоставила возможность программистам, знакомым с парадигмой *MapReduce*, писать пользовательские функции *map* и *reduce* и отправлять их в механизм параллельного потока данных *Greenplum* для обработки. Система *GPDB* взяла заботу о распространении входных данных, выполнении программы на узлах кластера, возможных сбоях в работе и управлении требуемой связью между узлами. Для обработки функций *map* и *reduce*, они должны быть определены в специально отформатированном документе *Greenplum MapReduce*, который затем передавался в клиентскую программу *Greenplum MapReduce* (*gpmapreduce*) для выполнения в среде *MPP* БД *Greenplum* [114].

Hadoop в те годы быстро стал предпочтительным решением для аналитики *Big Data* при работе с неструктурированной информацией, но из-за нехватки навыков и большого бремени программного обеспечения посчитали *Hadoop* дорогостоящим и неприемлемым для промышленной реализации. *Hadoop* — это проект *Apache Software Foundation*, предоставляющий среду для хранения данных и платформу обработки с двумя основными компонентами: распределенной файловой системой *HDFS* для хранения и *Hadoop MapReduce* для программирования заданий параллельной обработки и вычислений [115]. Такие компании, как *Cloudera, Hortonworks* и *MapR* предоставляют коммерческие услуги по поддержке инфраструктуры *Hadoop*, обеспечивающей законченную функционально развитую платформу для аналитики и управления данными благодаря его интеграции с рядом других актуальных проектов [116]. *EMC* и вошедшая в ее состав *Greenplum*, крупнейшие поставщики решений для хранения и обработки данных, также продвигают свой собственный дистрибутив *Hadoop*. По-видимому, этим и объясняется присутствие в технической документации, например, на *GPDB v*.4.3.6.1 раздела о совместимости реализаций *Hadoop* (табл. 1) от разных производителей [117]. Первый релиз (*GPDB v*.4.3) продукта содержал дистрибутив *Hadoop*

Таблица 1. **Совместимость реализаций *Hadoop***

Hadoop Distribution	***Version***	***gp_hadoop_target_version***
Pivotal HD	*Pivotal HD* 3.0	gphd-3.0
	Pivotal HD 2.0, 2.1 *Pivotal HD* 1.0 [1]	gphd-2.0
Greenplum HD	*Greenplum HD* 1.2 *Greenplum HD* 1.1	gphd-1.2 gphd-1.1 (*default*)
Cloudera	*CDH* 5.2, 5.3	cdh4.1
	CDH 5.0, 5.1	cdh4.1
	CDH 4.1[2] – *CDH* 4.7	cdh4.1
Hortonworks Data Platform	*HDP* 2.1, 2.2	hdp2
MapR[3]	*MapR* 4.*x*	gpmr-1.2
	MapR 1.*x*, 2.*x*, 3.*x*	gpmr-1.0
Apache Hadoop	2.*x*	hadoop2

Notes:

1. *Pivotal HD 1.0 is a distribution of Hadoop 2.0*
2. *For CDH 4.1, only CDH4 with MRv1 is supported*
3. *MapR requires the MapR client*

Pivotal HD, *Greenplum HB*, *Cloudera*, *Greenplum MR*.

Стратегия *Hadoop* от *EMC* была примерно такова: зная не понаслышке о недостатках текущей версии распределенной файловой системы *HDFS*, *EMC* захотела создать уровень хранения, который улучшил бы *HDFS* с точки зрения производительности, доступности и простоты использования. Но *EMC* обратила внимание на качественный продукт компании *MapR* [118–120], не желая затрачивать свой инженерный опыт, время и усилия на улучшение *HDFS*. *MapR Technologies* выпустила большой набор инструментария для данных, основанный на *Apache Hadoop*, с собственной альтернативой хранилищу *HDFS* — систему *MapR* (*MapR-FS*), представляющую собой кластерную файловую систему, обеспечивающую высокопроизводительное хранилище корпоративного уровня для больших данных. Программное обеспечение имеет как коммерческую, так и бесплатную версию. Первое из них содержит репликацию, моментальные снимки и зеркалирование файлов данных, восстановление мастера *MapReduce* (*Job Tracker*) [115] и коммерческую поддержку. Оно также содержит средство для перезапуска главного процесса *JobTracker* в течение нескольких секунд после сбоя и обеспечивает инициализацию подчиненных процессов *TaskTrackers* для повторного подключения. Возможная задержка в этом случае быть при выполнении заданий, но выполняемые работы будут продолжены и завершены, а не прерваны как в хранилище *Apache Hadoop*. В случае сбоя мастер-процесса, другая реплика прозрачно и немедленно берет на себя задачу обслуживания без прерывания процесса. *MapR* также поддерживает сжатие на уровне файловой системы и делает множественные копии метаданных в кластере для обеспечения доступности. Он распространяет метаданные по узлам (не требует их хранения в ОЗУ), позволяя одному кластеру поддерживать их значительное количество. Это отличается от *HDFS*, сохраняющей все метаданные файлов в памяти одной машины — *Name Node*.

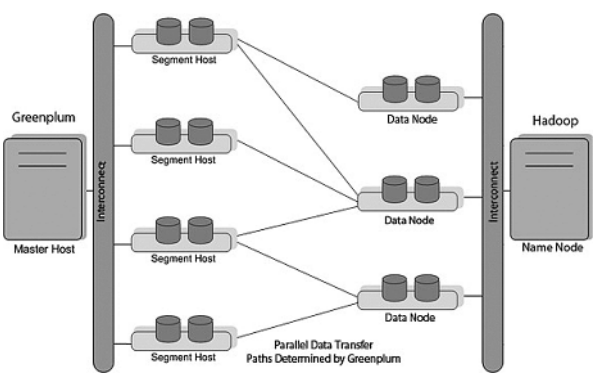

***Рис.* 19**

EMC использовала технологию, разработанную компанией *MapR*, для платформы аналитики *Hadoop* в продуктах *Greenplum*. По заявлению *MapR* [118], технология, которую она разработала на вершине *Hadoop*, поддерживает моментальные снимки для защиты и восстановления данных, исключает отдельные точки отказа и повышает производительность базовой инфраструктуры *Hadoop* в два-пять раз в сравнении со стандартным дистрибутивом *Apache Hadoop*.

Что касается компаний *EMC* и *Cloudera*, то они используют разные подходы к улучшению качества *HDFS*. Не привязывая себя к проекту *Apache Hadoop*, *EMC* может решать свои задачи, используя высокую доступность и производительность, а также расширенные функции *MapR*, такие как зеркалирование и репликация. *Cloudera*, являясь основным вкладчиком *Apache Hadoop*, будет включать изменения в архитектуру и функции *HDFS*, которые официально примет *Apache* [120]. Дистрибутив *Hadoop* от *EMC* не содержит официальной версии кода *Apache*, он основан на коде *Hadoop* от *Facebook* (*sub req'd*)[6] , который был оптимизирован для масштабируемости и многосайтового развертывания [119].

Интеграция платформ Greenplum и Hadoop. Внешние таблицы в сочетании с программой параллельного распределения файлов

[6] *Sub requirement* — отдельные компоненты, составляющие единое требование; конкретные детали того, что необходимо для выполнения этого требования (*Source: tada.uic.edu.*).

Greenplum (*gpfdist*) обеспечивают параллелизм, используя ресурсы всех сегментов *GPDB* при загрузке или выгрузке данных. БД *Greenplum* также использует параллельную архитектуру *Hadoop* для эффективного чтения и записи файлов данных из таблиц распределенной файловой системы *HDFS* посредством протокола *gphdfs*. Обмен данными между платформами *Hadoop* и *Greenplum* возможен посредством записываемых внешних таблиц (*Writable External Table*[7]), которые выводят данные в файлы. Когда БД *Greenplum* устанавливает связь с файлами *HDFS*, данные считываются параллельно с узлов *HDFS* в сегменты *GPDB* для обработки. БД *Greenplum* определяет соединения между сегментами и узлами. Внешняя таблица, расположенная на распределенной файловой системе *Hadoop*, приведена на рис. 19.

Разграниченный текст (*TEXT*) — единственный формат, допустимый в *GPDB v*. 4.1 для чтения и записи внешних таблиц файлов *HDFS* [91], в старших версиях БД *Greenplum* дополнительно реализованы пользовательские форматы (двоичные, *Pig*, *Hive* и т.д.) [121, 122]. Так, в *GPDB v*. 4.3.6.1 [123] реализовано создание и поддержка внешних таблиц для форматов файлов *Avro* и *Parquet* в системе *Hadoop* (*HDFS*) с использованием протокола *gphdfs* для чтения и записи файлов. Файл *Avro* хранит схему как определение данных, так и данные вместе в одном файле, что облегчает динамическое понимание информации, хранящейся в нем, программам. Схема *Avro* находится в формате *JSON*, данные — в двоичном формате, что делает его компактным и эффективным. Файл *Parquet* предназначен для использования сжатого, эффективного представления колоночных данных для проектов в экосистеме *Hadoop*. Он поддерживает сложные вложенные структуры данных и использует алгоритмы чередования и сборки *Dremel*, применяемые *Google* при интерактивном анализе наборов данных десятков-сотен терабайт за считанные секунды на основе *MapReduce* [124, 125]. *Parquet* поддерживает очень эффективную компрессию и кодирование, позволяет использовать схемы сжатия для каждого столбца и поддерживает добавление большего количества кодировок по мере их создания и реализации.

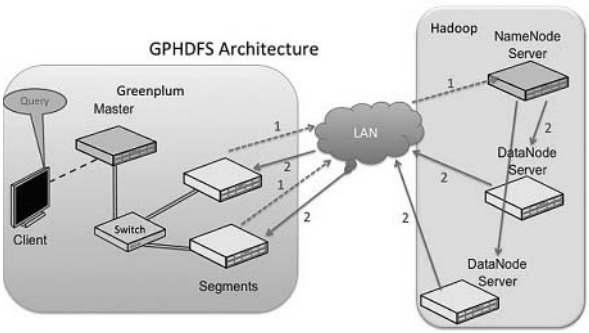

***Рис.* 20**

Одной из особенностей версии БД *Greenplum* 4.2 является использование файловой системы *Hadoop* (*HDFS*) для создания внешних таблиц, что позволяет избежать значительных затрат времени загрузки данных [91, 126]. В БД включена возможность предоставления привилегий для протокола *gphdfs* владельцу внешних таблиц [121 с.85], необходимых для их создания, которые обращаются к файлам на *HDFS*. Кроме того, предоставлен программный соединитель *Hadoop Connector*, обеспечивающий высокопроизводительный параллельный импорт и экспорт сжатых и несжатых данных из кластеров *Hadoop*. Чтобы дополнительно оптимизировать потребление ресурсов во время загрузки, данные, пользовательского формата в *Hadoop*, теперь могут быть преобразованы в формат *GPDB* с использованием *MapReduce*, а затем импортированы в нее. Это обеспечивает значительно более эффективный и гибкий обмен данными между *Hadoop* и БД *Greenplum* [127]. Последняя, по заявлению разработчика, поддерживает параллельную обработку *SQL* и *MapReduce* при объемах данных, варьируемых от сотен гигабайт, от десят-

[7] Записываемые внешние таблицы, посредством которых данные выводятся в файлы, используют программу параллельного файлового сервера *Greenplum* (*spfdist*) или интерфейс распределенной файловой системы *Hadoop* (*gphdfs*). Загрузить и записать пользовательские данные без *HDFS* можно, воспользовавшись поддержкой БД *Greenplum* и форматами импорта и экспорта данных *TEXT* и *CSV*.

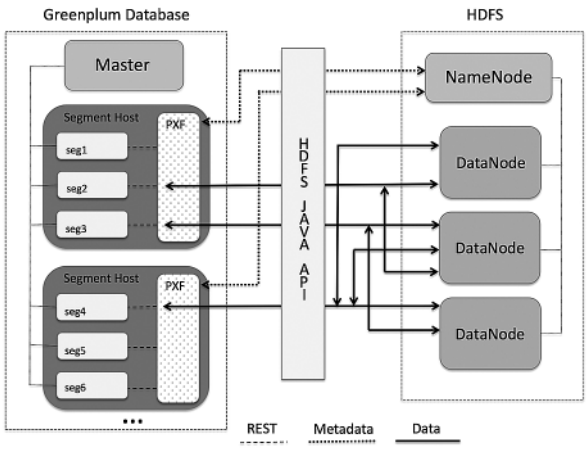

***Рис.* 21**

ков до сотен терабайт, до нескольких петабайт. Рис. 20 иллюстрирует процесс передачи файлов данных с кластера *Hadoop* в БД *Greenplum* по инициированному ею запросе (1) на чтение. Запрос (1) поступает на *NameNode* (*Hadoop*), который управляет доступом к запрошенным файловым блокам данных. *NameNode* отправляет (2) инструкции *DataNode* для предоставления блоков файлов в БД *Greenplum*.

Вместе с тем, нельзя не согласиться с мнением [128], что в настоящее время большинство аналитических баз данных и приборов интегрируются с *Hadoop*, но по-прежнему страдают от задержки отправки данных по сети из кластеров *Hadoop* в базу данных и обратно.

Доработка *Greenplum* параллельного интерфейса произведена в новом релизе [129] — это платформа расширения БД *Greenplum* (*PXF*) — фреймворк, позволяющий обмениваться данными со сторонними гетерогенными системами. Взаимодействие *PXF* с *HDFS* поясняется рис. 21. На нем БД *Greenplum* представлена главным сервером (*Master Host*) и несколькими серверами сегментов (*Segment Host*). Фреймворк *PXF* состоит из протокола БД *Greenplum*, клиентской библиотеки *C*, отвечающей интегрируемым хранилищам данных, и службы *Java*. Написанное расширение на *Java* представляет собой отдельный процесс *JVM PXF* на каждом *segment host* кластера *Greenplum*. Этот длительный процесс, одновременно обслуживающий множество очередей запросов и называемым агентом *PXF*, порождает поток для каждого сегмента — экземпляра базы данных (*segment instance*) на *segment host*. *PXF*, с одной стороны, обменивается данными параллельно с *segment instances* через *REST API*, с другой — агенты *PXF* на *segment hosts* параллельно взаимодействуют с платформой *HDFS*.

Имеется поддержка основных сервисов стека *Hadoop* и параллельная выгрузка данных из сторонних СУБД через *JDBC*. Расширение *PXF* предоставляет коннекторы для хранилищ данных *Hadoop, Hive* и *HBase*, которые определяют профили, поддерживающие форматы: *Binary* и *TEXT*, *Avro*, *JSON*, *RCFile*, *Parquet*, *SequenceFile* и *ORC*. *PXF*-коннекторы включают клиенты используемых хранилищ данных на каждом *segment host* БД *Greenplum* [130].

Расширение *PXF* реализует протокол с именем *pxf*, используемый для создания внешней таблицы, которая ссылается на данные в интегрируемом хранилище. После того, как расширение *PXF* зарегистрировано в базах данных, с которыми планируется использовать фреймворк для доступа к внешним данным, назначены привилегии для протокола *pxf* тем пользователям/ролям, которым требуется доступ, используется команда для создания внешней таблицы. При выполнении (пользователем или приложением) запроса на данные в *HDFS*, *Master Host* отправляет его всем серверам сегментов. Каждый *segment instance* связывается с агентом *PXF*, запущенным на его *segment host*, и когда агент *PXF* получает запрос от *segment instance*, он вызывает *API Java HDFS* для получения информации метаданных из *NameNode* для файла *HDFS* и предоставляет ее в экземпляр сегмента. Экземпляр сегмента использует свою базу данных g*p_segment_id Greenplum* и информацию *File Block*[8], содержащуюся в ме-

[8] *FCB* (*File Control Block*) — это внутренняя структура файловой системы, используемая в *DOS* для доступа к файлам на диске. Блок *FCB* содержит информацию о имени диска, имени и типе файла и другой информации, которая требуется системе при доступе или создании файла — *https://www.webopedia.com/TERM/F/File_Control_Block_FCB.html*. В *Hadoop* при помещении файла на *HDFS*, он разбивается на блоки,

таданных, для присвоения запросу определенной части данных. Затем экземпляр сегмента информирует агента *PXF* о необходимости чтения отобранных данных, которые могут быть на одном или нескольких *HDFS DataNodes*. Агент *PXF* вызывает *API HDFS Java* для чтения данных и доставки их на *segment instance*.

Расширение *PXF* более значимо для интеграции, в сравнении с предыдущими решениями, так как исключает сетевой компонент при обмене, имеет большую гибкость и возможность подключать сторонние системы, написав свой коннектор. В [131] приведены соображения о возможности интеграции БД *Greenplum* со сторонними СУБД через *JDBC*.

Отличия ПО интегрированных систем EMC Greenplum HD Enterprise и Community с Hadoop

Предприятия предъявляют чрезвычайно высокие требования к системам обработки, когда дело доходит до производственных данных. Две эти наработки в части реорганизации *Hadoop* для обеспечения высокой надежности и устойчивости в случае отказов его отдельных точек (*Name Node* и *Job Tracker*), доступности и разработки приемлемого механизма обмена данными между платформами в виде внешних таблиц, подобно симбиозу партнеров позволили достичь функциональности хранения и обработки данных в решениях *Community* и *Enterprise*. Отличие этих двух модификаций систем состоит в том, что *EMC Greenplum HD Community Edition* использует бесплатную, с открытым исходным кодом, версию распределенной файловой системы *MapR-FS M3 Edition*, основанную на ветке кода *Apache Hadoop* 0.20.1. — оптимизированной версии *Facebook Hadoop. Community Edition* (*CE*) полностью сертифицирована на совместимость с *open source* и поддерживает стек *Apache Hadoop*, состоящий из распределенной файловой системы *HDFS, MapReduce, HBase, Zookeeper* и *Hive*, а также обеспечивает отказоустойчивость для *Name Node*[9] в *HDFS* и мастера *Job Tracker* в *Hadoop MapReduce*. Этим обеспечивается высокая доступность кластера, позволяющая предотвратить потерянные задания, устранить перезагрузки и болезненные инциденты при сбоях.

В свободно распространяемое ПО *EMC Greenplum CE*[10] входят:

Greenplum Database CE — ведущий в отрасли продукт с *MPP* обработкой для крупномасштабной аналитики и хранилищ данных следующего поколения;

MADlib — библиотека аналитических алгоритмов с открытым исходным кодом, обеспечивающая параллельные реализации математических, статистических и машинных методов обучения для структурированных и неструктурированных данных;

Alpine Miner[11] представляет собой интуитивно понятный инструментарий для интеллектуального анализа данных, обеспечивающий быстрое *моделирование для оценки*, использует аналитику и специально предназначен для приложений с *Big Data*.

Базы данных *Greenplum* и *Greenplum HD* являются дополняющими технологиями, которые в совокупности предоставляют решение для анализа структурированных и неструктурированных данных. При совместном использовании аналитики могут выполнять сложные интерактивные исследования, используя воз-

размещаемые на *DataNodes* — последние хранят блоки, а не файлы. Метаданные находятся на *Name Node*, они содержат файл в *Block mapping,* запоминают расположение блоков в *DataNodes,* активные узлы и др. метаданные. Все они хранятся в памяти *NameNode* — *https://www.quora.com/What-does-the-term-metadata-mean-in-Hadoop*

[9] До *Hadoop* v2.0.0 *Name Node* было единой точкой отказа (*SPOF, Single Point Of Failure*) в кластере *HDFS*. С *Zookeeper* функция высокой доступности *HDFS* решает эту проблему, предоставляя возможность запуска двух избыточных *Name Node* в одном кластере в конфигурации *Active/Passive* с горячим резервом — https://hadoopecosystemtable.github.io/

[10] *EMC Greenplum Introduces Free Community Edition of ""Big Data"" Tools for Developers and Data Scientists* – https://www.emc.com/about/news/press/2011/20110201-02.htm

[11] *Alpine Miner Visual Modeler* — инструмент анализа независимого разработчика.

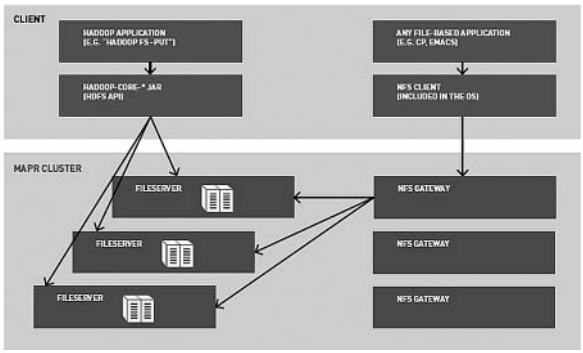

***Рис.* 22**

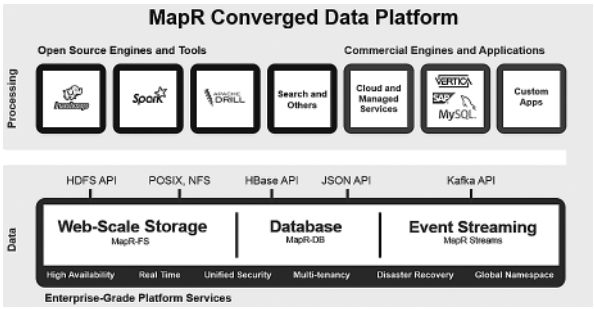

***Рис.* 23**

можности *Greenplum SQL* и расширенные аналитические функции по данным в *Hadoop* и/или в БД *Greenplum*. Доступ к данным и их перемещение между платформами упрощаются с помощью внешних таблиц, чтобы обеспечить быстрый и простой параллельный обмен данными.

EMC Greenplum HD Enterprise Edition — известна также как *Greenplum MR* — интегрирует дистрибутив *MapR M*5 *Hadoop* (обновленная версии *M*3 с высокой доступностью и функциями защиты данных), заменяя *HDFS* на собственную файловую систему *MapR-FS* [134]. Архитектура системы хранения, используемая *MapR-FS*, написана на *C/C* ++ и устраняет конкуренцию (*lock contention*), исключая влияние сборки мусора *Java* на производительность, что свойственно *Apache Hadoop*. *MapR-FS* представляет собой распределенную файловую систему произвольного чтения и записи, позволяющую приложениям одновременно считывать и записывать данные непосредственно в кластер. Напротив, распределенная файловая система *Hadoop* (*HDFS*) предназначена для записи только с добавлением[12] (*append-only*) и чтения только из закрытых файлов.

Протокол сетевой файловой системы (*Network File System, NFS*) [135] обеспечивает приложениям прямой доступ непосредственно к кластеру, позволяя передавать потоки данных в режиме реального времени. Стандартные приложения и инструменты напрямую обращаются к слою хранения *MapR-FS* с использованием протокола *NFS*. Устаревшие системы получают доступ к данным, а традиционные операции ввода-вывода файлов работают в обычной файловой системе *UNIX*. Удаленный клиент может монтировать кластер *MapR* для перемещения данных в кластер и из него.

Реализация MapR Direct Access NFS™. На каждом узле кластера *MapR* есть сервис *FileServer*, роль которого во многом сходна с *Data Node* в файловой системе *Hadoop HDFS*. Кроме того, в кластере может быть одна или несколько служб *NFS Gateway*[13]. Во многих развертываниях сервис работает на каждом узле кластера, наряду с *FileServer*. Запуск нескольких серверов *NFS Gateway* обеспечивают устойчивость к отказу. Виртуальный *IP*-адрес (*VIP*) доступа к *NFS* автоматически переносится в случае сбоя от одного сервиса *NFS Gateway* к другому, так что клиенты, подключившиеся к кластеру через *IP*, могут продолжить чтение и запись данных практически без задержек. В типичном развертывании для равномерного распределения клиентов между различными серверами шлюза *NFS* (т.е. *VIP*) используется простая система балансировки нагрузки, такая как циклический *DNS*. В кластере *MapR* служба шлюза *NFS*

[12] Когда проект *Apache Hadoop* создавался, он должен был использоваться для веб-сканирования и индексирования Интернета. Это требовало не более чем парадигмы с однократной записью (или только с добавлением — or *append-only*), которая легла в основу реализации *HDFS*. Веб-сканирование — чрезвычайно ограниченный случай среди всех применений, которые действуют сегодня для *Apache Hadoop* [*Get Real with Hadoop: Read-Write File System – https://mapr.com/blog/get-real-hadoop-read-write-file-system/*].

[13] *MapR Direct Access NFS* — https://mapr.com/resources/mapr-direct-access-nfs/

получает запросы от клиента и преобразует их в запросы *RPC*[14] к соответствующим службам *FileServer*. Взаимодействие с кластером *MapR* по протоколу *NFS* показано на рис. 22.

Поддержка случайных чтений и записи необходима для обеспечения прямого доступа *NFS* и, в более общем случае, к любому виду доступа для отличных от *Hadoop* приложений. Помимо прямого доступа, предоставляемого *NFS*, есть некоторые другие особенности *MapR-FS*. Кластер доступен как каталог в локальной файловой системе и приложения на клиентском узле работают по протоколу *NFS* с данными файловой системы *MapR* таким же образом, как при доступе к ним на локальном диске, используя его в режиме реального времени. Предоставляет также специальные функции для управления данными и высокой доступности (*HA*), такие как логические тома – способ группировки данных и применения политики во всем наборе данных, использующиеся для распространения метаданных в клас-тере и для резервного копирования данных; контейнеры – абстрактный объект, хранящий файлы и каталоги в *MapR-FS*, например, может содержать информацию о пространстве имен, фрагментах файлов или части таблицы для тома, к которому принадлежит контейнер; предоставляет услуги отслеживания местоположения каждого контейнера и др.

MapR [136] — это полный дистрибутив корпоративного класса для *Apache Hadoop*. Платформа конвертируемых данных (*Converged Data Platform*) *MapR* предоставляет единое решение для структурированных (таблиц) и неструктурированных данных (файлов), разработана для повышения надежности, производительности и простоты использования *Hadoop*. В дистрибутиве *MapR* представлен полный стек *Hadoop*, включающий файловую систему *MapR-FS*, систему управления базами данных *MapR-DB NoSQL*, потоки *MapR* и семейство проектов *Hadoop*. Высокоуровневое представление платформы конвергентных данных *MapR*, ее основные компоненты и поддерживаемые экосистемные проекты, а также средства взаимодействия с ними приведены на рис. 23. Взаимодействие с *Apache Hadoop, HDFS* и *MapReduce* происходит посредством *API MapR*. Архитектура *Hadoop* 2.*x* и *YARN* поддерживается *MapR* [115].

MapR предоставляет несколько уникальных функций, которые затрагивают общие проблемы с *Apache Hadoop* [136]:

- защита данных (*MapR Snapshots*) — поддерживает согласованные моментальные снимки для файлов и таблиц, зеркалирование и репликацию, в том числе и удаленную, для полного восстановления данных;
- аварийное восстановление (*Disaster Recovery*) — *MapR* обеспечивает бесперебойную работу кластера и службы аварийного восстановления с помощью настраиваемого зеркалирования, эффективно использующего ресурсы хранилища, процессора и пропускную способность кластера;
- масштабируемая архитектура без единой точки отказа (*SPOF*) — платформа конвергентных данных *MapR* обеспечивает высокую доступность для стека компонентов *Hadoop*. Кластеры *MapR*, в отличие от других дистрибутивов *Hadoop*, не используют *Name Node*, приводящего к сбоям кластера, и обеспечивают сохранение доаварийного состояния для *Job Tracker* в *Hadoop MapReduce* и механизма прямого доступа к кластеру *MapR* по протоколу *NFS* без какой-либо специальной настройки. В *Apache Hadoop*, с предоставлением возможности запуска двух избыточных *Name Node* (*HA*) в одном кластере, *Name Node HA* обеспечивает переход на другой ресурс, но без отката, ограничивая численность узлов и создавая сложные проблемы с конфигурацией. Высокая доступность *JobTracker* в *Apache Hadoop* не сохраняет состояние рабочих заданий. Отказоустойчивость для *JobTracker* связана с перезапуском всех незавершенных заданий и усложняет конфигурацию фреймворка;
- *производительность* — *MapR* использует индивидуальные модули ввода-вывода, коммутации, повторной синхронизации и администрирования. Эти архитектурные элементы

[14] *Remote Procedure Call* (*RPC*) – удаленный вызов процедур (RPC).

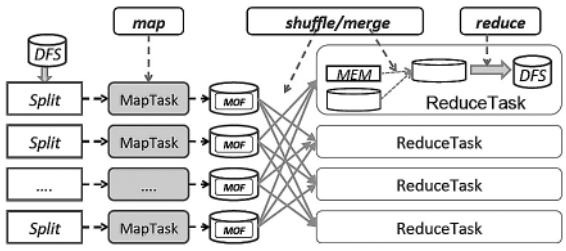

***Рис*.24**

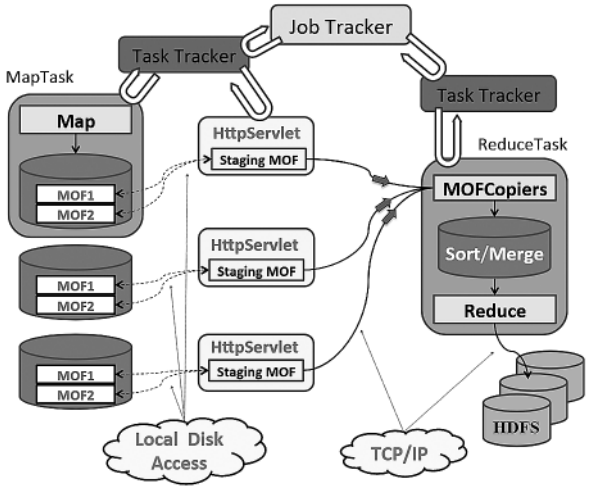

***Рис*.25**

позволяют кластерам *MapR* работать со скоростями, близкими к максимально допустимыми базовым оборудованием. Кроме того, технология *DirectShuffle*, построенная *MapR* на основе ряда улучшений привнесенных в структуру *MapReduce*, использует преимущества *MapR-FS* для обеспечения высокой производительности и управляемости кластера, а механизм прямого доступа к кластеру по протоколу *NFS* упрощает прием и доступ к данным [136].

Высокая производительность и надежность выполнения задач *MapReduce* (*Vers*.1) достигается технологией *DirectShuffle* [137, 138]. В выпусках *Apache Hadoop* до версии 1.55.X [138] процесс [139–141], посредством которого промежуточные выводы обработки сегментов данных от *mappers* (*MOF, MapOutputFile*) переносятся на *reducers*, называется перетасовкой (*shuffling*). *MOFs* по завершению задачи *Map* сохраняются на локальных дисках. При этом выполняется их сортировка (*sorting*), необходимая для каждой отдельной задачи *Reduce*, и передача сегмента *MOF* в *reducer* в качестве входного сигнала. В свою очередь, фаза сортировки в *MapReduce* охватывает и слияние (*merger*). Промежуточные результаты, представляющие собой записи данных в виде пар ключ-значение (*key-value*), разделяются между *reducers* и сортируются автоматически по ключу. Каждый *reducer* получает все значения (*value*), связанные с одним и тем же ключом, т.е. в процессе сортировки *MOFs* также происходит их объединение в порядке соответствия ключей в записях данных. Фазы *shuffle* и *sort* в *Hadoop* происходят одновременно и выполняются *MapReduce Framework*. Результат задачи *Reduce* сохраняется в файловой системе *HDFS* (рис. 24 и 25).

Перетасовка и сортировка в *Hadoop MapReduce* вообще не выполняются, если указываются нулевые (*zero*) *reducers* (*setNumReduceTasks* (0).

Главный процесс *JobTracker* координирует задания в кластере (рис. 25) и назначает процессы *TaskTracker* на узлы для координации выполнения задач. В каждом *TaskTracker* присутствует встроенный *HttpServer*, который создает множественные *HttpServlets* для каждого запроса *reducer* на получение результатов обработки сегментов данных от *mappers*. *MOFCopier* используется для сбора этих результатов, предназначенных задачам *Reducers*. Промежуточные результаты считываются с *Local Disks* и транспортируются *HttpServlets* к *MOFCopier* посредством *HTTP* запросов. Модель сетевого взаимодействия, используемая для пересылки данных в *Apache Hadoop*, приведена на рисунке 26. Используемые стандартные средства языка *Java* для работы с *HTTP* запросами и сокетами являются узким местом в работе всей системы, поскольку использование *Java Virtual Machine* для доступа и перемещения данных при помощи потоков *FileInputStream* работает гораздо медленнее, чем реализации, например, на языке *C* [142]. Архитектура системы хранения файловой системы *MapR-FS* написана на *C/C++* и вместо *HTTP* использует соединения удаленного вызова процедур *RPC*.

Кроме того, в фазе *Shuffle* перетасовки промежуточных результатов задачи *MapReduce* возникает проблема роста сетевой нагрузки в *Hadoop* из-за большого объема сетевого обмена. Перетасовка данных в пяти процентах больших заданий может потреблять более 100 процентов пропускной способности сети кластера, и, что еще хуже, производительность *Hadoop* с увеличением среднего размера данных, растет нелинейно [142]. Чрезмерная нагрузка на пропускную способность сети может быстро насытить сетевые связи тех машин, которые участвуют в фазе *reduce* [143]. Перетасовка данных в фазе *Shuffle*, по существу, становится доминирующим источником сетевого трафика и узким местом в производительности в *Hadoop*.

MapR обеспечивает повышение производительности в фазе перетасовки *MapReduce*. Эта фаза включает в себя изготовление многих копий и значительное число координаций между узлами в кластере, что приводит к чрезмерному росту сетевой нагрузки в *Hadoop*. Другое, поскольку *HDFS* находится поверх файловой системы *Linux*, большое число операций ввода/вывода (*I/O*) снижает производительность клстера. Кроме того, вывод результатов *mappers* на локальные диски создает конкуренцию за дисковое пространство между локальными и распределенным хранилищем.

MapR, вместо записи промежуточных данных на локальные диски, контролируемые операционной системой, обращается непосредственно к томам *MapR-FS* на локальных узлах. Это повышает производительность *Hadoop* и снижает спрос на локальное дисковое пространство. В *MapR* любые закачанные (*spilled*) данные хранятся в *MapR-FS*, что делает их доступными напрямую. Прямое перемещение использует базовый уровень хранения и его возможности (*MapR-FS* написана на *C/C++* и содержит протокол *NFS*). Перетасовка в *MapR-FS* намного быстрее, потому что *MapR* использует высоко оптимизированные эффективные соединения удаленного вызова процедур (*RPC*) для передачи данных непосредственно из любой точки сети [134, 135, 137].

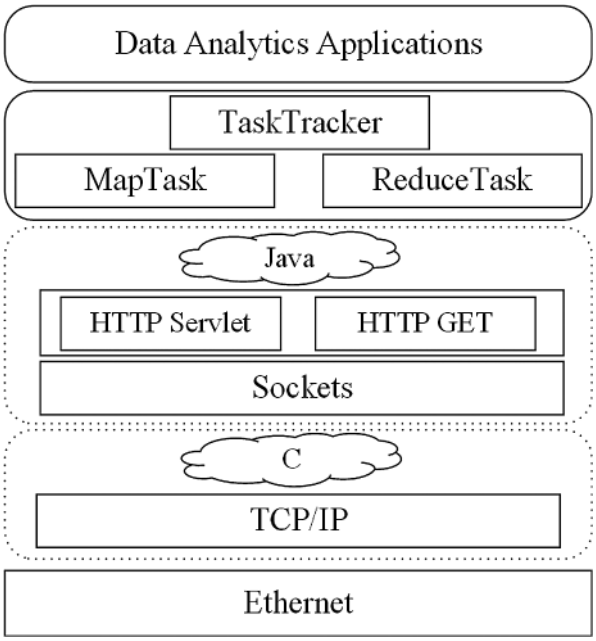

Рис. 26

Возможно использование нескольких сетевых адаптеров посредством соединения на уровне *RPC* процедур. Для сравнения, операция перетасовки в других дистрибутивах *Hadoop* может использовать только один сетевой адаптер (теоретически, можно применить транкирование портов, но прирост производительности будет минимальным в сравнении с балансировкой нагрузки на уровне *RPC* в *MapR*) [137, 138].

Особенности реализации *Direct Shuffle* на *MapR* 6.0 *Platform* с *YARN* приведены в [144]. Подход *MapR* к *Direct Shuffle* привел к ускоренному от двух до пяти раз выполнению заданий *MapReduce* [118, 138]. Производительность *Greenplum HD EE* в сравнении со стандартной версией *Apache Hadoop*, по оценкам *EMC*, также улучшена на эту величину [145]. Слияние технологий *EMC Greenplum DB* и предварительно интегрированного, проверенного и упрочненного дистрибутива *MapR Technologies* для стека *Apache Hadoop*, осуществляет совместную обработку разных типов данных в едином, бесшовном решении. *Greenplum HD EE* полностью совместима с *Apache Hadoop Interface*, так как дистрибутив *MapR Hadoop* позволяют напрямую обмениваться данными через протокол

NFS прямого доступа. *Hadoop* в этом комбинированном решении рассматривается как ключевая платформа обработки данных в дополнение к существующим инфраструктурам, и реализует в отрасли законченную платформу для аналитики больших данных [145, 146]. Сюда же входит *EMC Greenplum HD Data Computing Appliance* (*DCA*) — специализированный вычислительный комплекс, включающий *EMC Greenplum HD Enterprise Edition* и весь набор необходимых стандартных аппаратных средств. Основное отличие *DCA* — динамичность, при которой в режиме, близком к реальному времени, анализируются данные разных типов, в том числе и внешних, источников [113].

Обновленное решение (сопоставьте решение[15] о роли *MapReduce*) для анализа массивов больших и, как правило, неструктурированных данных, привносит интегрированная среда *Apache Hadoop MapR Technologies*, включающая файловую систему *MapR-FS, MapReduce, Hive, Pig, HBase* [115], *ZooKeeper* и *MapR Heatmap*. *Zookeeper* — это система для координации распределенных процессов. *MapR Heatmap* положена в основу *Control System Greenplum HD EE*, обеспечивает статус оборудования и управление с помощью комплексного пользовательского интерфейса (*user interface, UI*), включающего в себя тепловую карту (*Heatmap*), которая отображает жизнеспособность кластера при работе. Графические и программные интерфейсы предназначены для масштабирования кластера.

Интеграция системы *NAS EMC Isilon с Hadoop*. Еще одно, подобно представленному выше, предложение интеграции горизонтально масштабируемой системы *NAS EMC Isilon* с распределенной файловой системой *Hadoop*. *NAS* (*Network Attached Storage*) — сетевая система хранения данных на уровне файлов, выполненная на специализированном серверном оборудовании и упрощенной операционной системе. Объединение *EMC Greenplum HD DCA CE* [147–149] и *NAS Isilon* в одном решении позволило рассматривать его как высокоэффективную и гибкую экосистему хранения и анализа больших данных.

EMC включила распределенную файловую систему *Hadoop* (*HDFS*) в качестве встроенного протокола, поддерживаемого операционной системой *OneFS® Isilon*, в дополнение к поддержке сетевой файловой системы (*NFS*) — единой файловой системы Интернета (*Common Internet File System, CIFS*). Это позволило *Isilon* предоставлять базовый уровень хранения, а также общий пул хранения для *Hadoop* и других систем. В реализации *Hadoop* на кластере *Isilon Date Storage* (рис. 27) масштабируемой платформы *NAS*, *OneFS* по функциональности соответствует *HDFS*, т.е. служит файловой системой для *Hadoop*. Файловая система *HDFS* в *Isilon* поддерживается как протокол, используемый вычислительными ресурсами *Hadoop* (*Hadoop Compute Resources*) для доступа к данным на уровне хранения *HDFS*. Клиенты

[15] Речь о *MapReduce*, которая была принята как метод высокопроизводительного анализа данных лидерами Интернета, такими как *Google* и *Yahoo*. *MapReduce* позволила программистам запускать аналитику с петабайтными наборами данных, которые хранятся как в БД *Greenplum*, так и за ее пределами. *Greenplum MapReduce* приносит преимущества надежности и известности реляционной базы данных набирающей силу модели программирования [92].

Таблица 2. ***HAWQ:*** **«драгоценности короны»** ***Greenplum***

▪ Высокопроизводительная обработка запросов с *Queru Optimizer* – масштабируемость диапазона петабайт, – интерактивная и подлинная поддержка *ANSI SQL*, – программируемая аналитика
▪ Услуги корпоративного класса базы данных *Enterprise-Class* – колоночное хранение данных и индексирование, – управление рабочей нагрузкой
▪ Комплексное управление данными – загрузка посредством технологии потоковой передачи данных *SG Streaming*, – многоуровневое партиционирование, – инструменты сторонних разработчиков и открытые клиентские интерфейсы

Hadoop могут обращаться к данным, хранимым в кластере *Isilon*, подключаясь к его узлам по протоколу *HDFS* [150].

Реализация *Hadoop* на хранилище *Isilon Date Storage* с *OC OneFS* имеет отличия. Все узлы кластера *Isilon* содержат *NameNode* и *DataNode*, что исключает маршрутизацию одним *NameNode*. *OneFS*, позволяя всем узлам кластера хранения данных *Isilon* стать, по сути, *NameNodes*, устраняет риск последствий для него единой точки отказа *Hadoop*. Это значительно повышает отказоустойчивость среды *Hadoop*. *Framework Hadoop MapReduce* и его компоненты не устанавливаются. Клиенты *Hadoop Compute Resources* могут подключаться к тем узлам кластера *Isilon*, которые функционируют как *NameNode* (см. рис. 27).

Функциональность уровня хранения выполняется *OneFS* на кластере *Isilon*. Уровень хранения в *OneFS* реализован в качестве встроенного нативного *HDFS*-протокола облегченного (*lightweight*) уровня, используемого между кластерами *Hadoop Compute Resources* и *Isilon*. Клиенты вычислительного кластера *Hadoop*, используя *HDFS* как протокол передачи данных, подключаются к хранилищу *Isilon*. В дополнение к *HDFS* клиенты вычислительного кластера *Hadoop* могут подключаться к кластеру *Isilon* по любому стандартному протоколу, поддерживаемому *OneFS*, например *CIFS, FTP, HTTP, NFS, SMB. Isilon OneFS* — нестандартная реализация *HDFS*, которая позволяет использовать *multi-protocol* доступа. *Isilon* предоставляет альтернативную систему хранения для *HDFS*, поддерживая службы *HDFS* с функциями управления данными корпоративного уровня [150].

Последние наработки (*Greenplum MR, EMC Greenplum HD DCA CE и NAS Isilon, Greenplum Chorus*) послужили дальнейшему развитию платформы *DCA — EMC® Greenplum® DCA Unified Analytics Platform* (*UAP*) *Edition*, позволяющей совместно анализировать структурированные и неструктурированные данные в одном интегрированном устройстве. Унифицированная аналитическая платформа, анонсированная в конце 2011 г., соединила в себе три решения

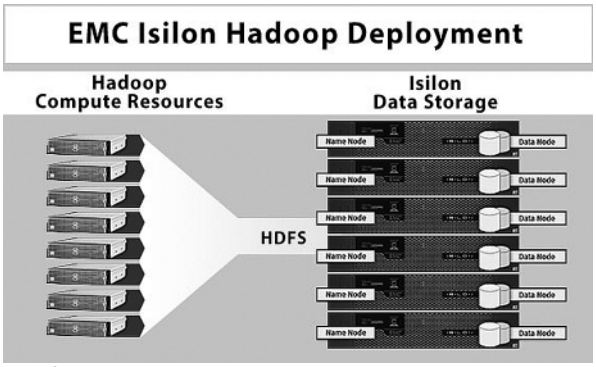

Рис. 27

для поддержки аналитики больших данных: *Greenplum DB* (структурированные данные); *Greenplum HD* (неструктурированные данные); *Greenplum Chorus* — единая совместная платформа исследования данных (*Data Science*). Основная цель создания *UAP* — максимальная интеграция результатов аналитики больших данных с бизнес-процессами. Потребители могут воспользоваться новым *DCA* для повышения производительности БД *Greenplum* для лучшей в своем классе обработки *SQL* и загрузки данных, а также использовать инновационные возможности распределения *Hadoop Greenplum* (*GPHD*). Совместимость анализа структурированных данных достигается встроенной поддержкой компонентов экосистемы *Hadoop: Hive, HBase, Pig и Mahout*.

Корпорация *EMC* улучшила свое решение *DCA*[16] путем его редизайна оптимизированными для аналитики масштабируемыми системами, используемыми для статистического анализа, интеллектуального моделирования и машинного обучения. *DCA UAP* интегрирует масштабируемое *NAS*-хранилище *EMC Isilon* со встроенной поддержкой системы *HDFS* и, в силу архитектурных особенностей, обеспечивает высокоскоростной прием данных, надежную и эффективную их защиту с функцией создания моментальных снимков файловой системы, зеркалирования, резервного копирования, восстановления, реплицирования и автоматической балансировки. Гарантируется

[16] *EMC Announces Enhanced Greenplum Big Data Analytics Appliance* — https://www.emc.com/about/news/press/2013/ 20130130-02.htm

безопасность и целостность данных в течение всего жизненного цикла: от их создания до записи в архив. Интеграция *DCA* с системами хранения данных, использующих технологию дедупликации *EMC Data Domain*, обеспечивает резервное копирование и восстановление для модулей БД *Greenplum* с услугами широкомасштабной репликации для расширенного аварийного восстановления.

Самообслуживание данными (*self-service data*), включая расширенную поддержку партнеров, таких как *SAS* и *Informatica* [154], и интерфейсы для специалистов в области аналитики, — предоставляет возможность пользователям обнаруживать и использовать любые данные предприятия в целях улучшения бизнеса.

Проект *EMC Pivotal Greenplum HAWQ*. *Apache HAWQ* (произносится как *hawk*) — базовый *SQL Apache Hadoop* [155]. Проект *HAWQ*, уровень базы данных *SQL*, расположенный поверх распределенной файловой системы *Hadoop* (*HDFS*), – часть нового пакета *Hadoop*, который *EMC* называет *Pivotal Hadoop Distribution* или кратко *Pivotal HD* [156]. Примерно такое же определение *HAWQ* от *Quora* [157] — это порт базы данных *MPP Greenplum*, с планировщиком и *PQO* (*Pivotal Query Optimizer*) — оптимизатором запросов для больших данных, теперь использующей *HDFS* для своего уровня хранения. Тем самым подчеркивается, вклад компании *Greenplum* в проект *HAWQ*: это наилучшие из достижений коллектива *Greenplum* (табл. 2), опирающееся на десять лет основополагающих исследований и разработок, приведших к созданию масштабируемой параллельной реляционной БД *Greenplum* и сделавшее ее претендентом на альтернативу *Teradata, IBM, Oracle* и других баз данных. То, что адаптация БД для *HDFS* выполнялась в течение более двух лет крупнейшей командой разработчиков *Hadoop* на планете (свыше 300 человек), дает основание утверждать: "*was all-in on Hadoop*"— все включено в *Hadoop*. Эти работы проводились, потому что компания полагала: *Hadoop* станет основой новой «текстуры, ткани данных» — *data fabric*[17] [156]. Термин *enterprise data fabric* (*EDF*) обозначает высокопроизводительный интерфейс (коммутирующая текстура или матрица) между системами хранения данных и использующими их приложениями.

HAWQ считывает данные и записывает данные в *HDFS* изначально. *HAWQ* — это собственный механизм запросов *SQL Hadoop*, который сочетает в себе ключевые технологические преимущества базы данных *MPP* с масштабируемостью и удобством *Hadoop*. *HAWQ* предоставляет пользователям полный *SQL*-интерфейс, совместимый со стандартами: *SQL*-92, *SQL*-99, *SQL*-2003 и др., расширением *OLAP*. Чрезвычайно высокая производительность – во много раз быстрее, чем другие известные механизмы *Hadoop SQL*. В сравнении с пакетно-ориентированными запросами, запущенными на кластере *Hadoop*, комбинация *HDFS* и *HAWQ* показывает всюду улучшения производительности от 10*X* до 600*X*, согласно *EMC*. Именно это превращает пакетную систему в интерактивную (недостижимое для мейнфреймов еще в 60-х и 70-х годах XX ст.) — необходимое условие того, чтобы *Hadoop* стал частью инструментария ИТ [156].

Высокоуровневое архитектурное представление *HAWQ* приведено на рис. 28 [158]. В типичном развертывании каждый подчиненный узел имеет один физический сегмент *HAWQ*, установленный узел данных *DataNode HDFS* и *NodeManager*. Мастера для *HAWQ, HDFS* и *YARN* размещаются на отдельных узлах. В *HAWQ* сегменты — это единицы, которые обрабатывают данные одновременно. На каждом хосте есть только один физический сегмент. Каждый сегмент может запустить много исполнителей запросов (*QE*) для каждого фрагмента запроса. Это заставляет один сегмент действовать как несколько виртуальных сегментов, что позволяет *HAWQ* лучше использовать все доступные

[17] *Data Fabric: Data Transport* — это видение будущего управления данными, архитектура для гибридного облака. Речь идёт не о конкретной технологии, а об экосистеме, стеке технологий, в которой пользователь может быстрее реагировать и внедрять новшества, перемещать данные и приложения в облачные сервисы и размещать рабочие нагрузки на наиболее подходящей платформе. Основные принципы изложены в — http://megatrade.ua/news/reviews/data-fabric-data-transport/

ресурсы. Виртуальный сегмент ведет себя как контейнер для *QE*. Каждый виртуальный сегмент имеет одно *QE* для каждого фрагмента запроса. Количество используемых виртуальных сегментов определяет степень параллелизма (*DOP*) запроса. Основной единицей параллелизма *HAWQ* является экземпляр сегмента.

Представление о программных компонентах дает рис. 29. *HAWQ* тесно интегрирован с *YARN* — структурой управления ресурсами *Hadoop*, в частности, ресурсами запросов. *HAWQ* кэширует контейнеры из *YARN* в пуле ресурсов, а затем управляет этими ресурсами локально, используя собственное тонкое управление ресурсами *HAWQ* для пользователей и групп. Чтобы выполнить запрос, *HAWQ* выделяет набор виртуальных сегментов в соответствии со стоимостью запроса, определениями очереди ресурсов, локальностью данных и текущим использованием ресурсов в системе. Затем запрос отправляется на соответствующие физические хосты, которые могут быть подмножеством узлов или всего кластера. Средство проверки ресурсов *HAWQ* на каждом узле отслеживает и контролирует ресурсы, используемые запросом в реальном времени, чтобы избежать нарушений в использовании ресурсов. Мастер *HAWQ* является точкой входа в систему. Он аутентифицирует клиентские соединения, обрабатывает входящие команды *SQL*, анализирует и оптимизирует запросы, распределяет рабочую нагрузку по сегментам, и координирует выполнение запроса. Запрос *SQL* в сегменты отправляется вместе со связанной информацией метаданных для обработки. Метаданные содержат *URL*-адрес *HDFS* для требуемой таблицы. Сегмент обращается к соответствующим данным, используя этот *URL*. Конечные пользователи взаимодействуют с *HAWQ* посредством мастера и могут подключаться к базе данных, используя клиентские программы, такие как *psql* или интерфейсы прикладного программирования (*API*).

В него входит глобальный системный каталог, содержащий в системных таблицах мета-данные о самой системе *HAWQ*. Служба каталога *HAWQ* хранит информацию об отно-

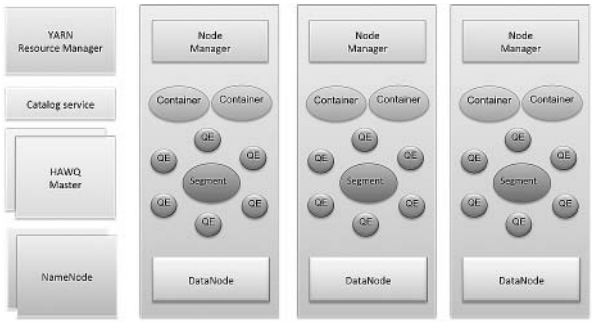

***Рис.* 28**

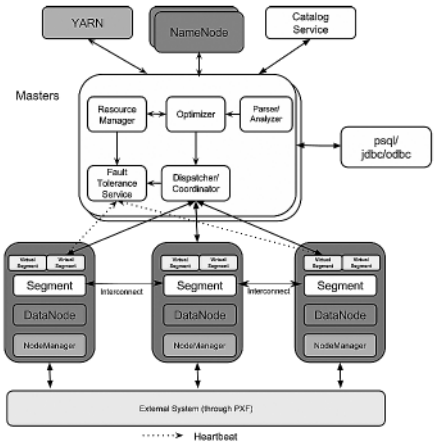

***Рис.* 29**

шениях, о безопасности и местонахождение файлов данных.

Служба отказоустойчивости *HAWQ* (*Fault Tolerance Service, FTS*) отвечает за обнаружение отказов сегмента и прием сигналов отклика от сегментов.

Диспетчер *HAWQ* отправляет планы запросов выбранному подмножеству сегментов и координирует выполнение запроса. Диспетчер и менеджер ресурсов *HAWQ* являются основными компонентами, отвечающими за динамическое планирование запросов и ресурсы, необходимые для их выполнения.

И в завершение обзора о разработках, инициатором которых был *Greenplum*, предоставим платформу *Pivotal и Greenplum* следующей генерации обработки данных (рис. 30).

Таким образом, имея весьма положительный отзыв[18] об аналитической *RDBMS MPP*

[18] *Greenplum is being open sourced. — Febr.* 18, 2015 – http://www.dbms2.com/2015/02/18/greenplum-is-being-open-sourced/

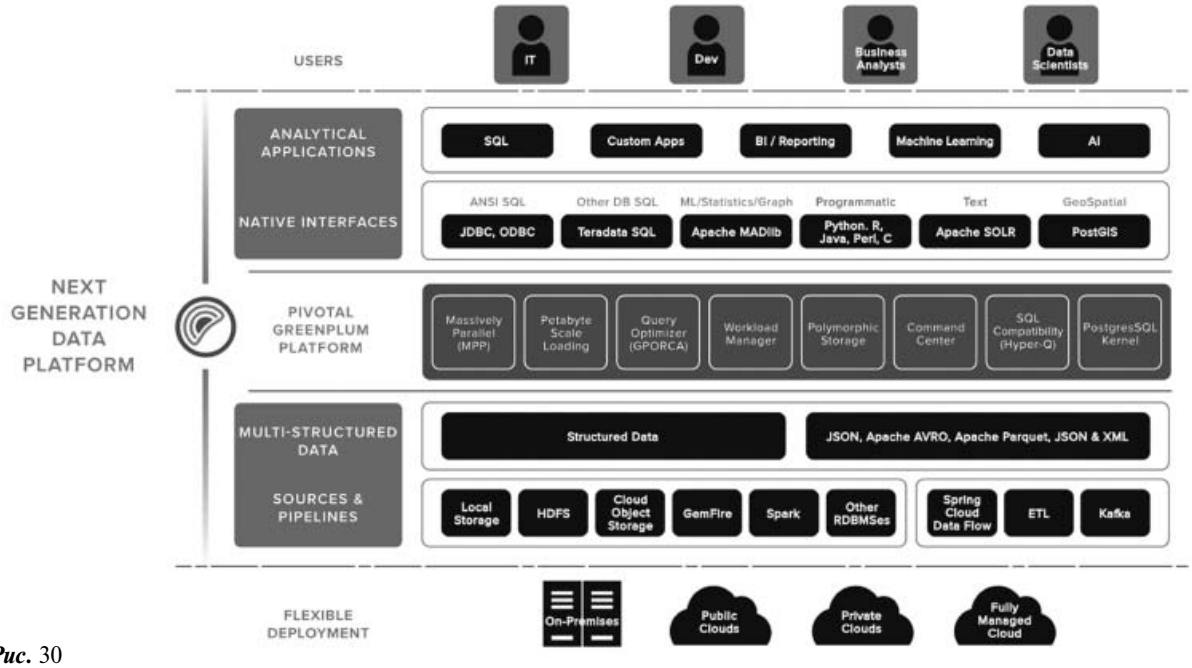

Рис. 30

с открытым исходным кодом *Greenplum*, обратился к проблемам *BigDate*. Реорганизация *Hadoop* для обеспечения высокой доступности, связанной с наличием отдельных точек отказа, замена файловой системы *HDFS* на *MapR*, чтобы ускорить работу *MapReduce* и обеспечить многомерную масштабируемость и др., позволили создать ряд интересных разработок.

Оптимизации производительности *Greenplum* и расширению его функциональности способствуют широкое использование компонентов экосистемы *Hadoop* (*Apache Hive, HBase, Flume, Kafka, Spark, Solr* и др., библиотек машинного обучения *MLLib, MADlib, Mahout*). Вместе с тем, постоянно увеличивающееся количество данных требует роста скорости их обработки. Сейчас есть альтернатива пакетному *MapReduce* [115] — *Apache Spark*, быстрая универсальная вычислительная среда для распределенной крупномасштабной обработки данных. Он производит большинство вычислений в памяти и вносит в экосистему *Hadoop* эффективные средства оперативной обработки, *Spark* — это интерактивно-аналитическая система, предназначенная для задач, использующих многопроходные обработки (машинное обучение, граф-анализ, углубленный анализ данных).

БД *Greenplum* поддерживает высокоскоростную параллельную передачу данных между кластером *Apache Spark* посредством *Pivotal Greenplum-Spark Connector*[19], что позволяет выполнять быструю аналитическую обработку в памяти, проводить интерактивный анализ, пакетную и потоковую *ETL*-обработку данных. Встроенная поддержка компонентов экосистемы *Hadoop* обеспечивает совместимость анализа структурированных данных.

PostGIS (рис. 30) — пространственное расширение БД *PostgreSQL*, позволяющее хранить и обрабатывать объекты географической информационной системы (ГИС) в базе данных. Геопространственные данные для последующих распределенных параллельных вычислений требуют своего разделения на части, партиции и классификации их по координатам измерений — пространственной индексации (геодезические сети, дерево квадрантов, *Grid-*

[19] *Overview of the Greenplum-Spark Connector* — https://greenplum-spark.docs.pivotal.io/

решетки и др.). Расширение *PostGIS* включает в себя поддержку пространственных индексов и функций для анализа и обработки объектов ГИС.

Сегодня эту нагрузку способны выполнять фреймворки: *GeoJinni* (прежде *SpatialHadoop*), *GeoMesa* и *GeoWave*. Последний задумывался как аналог *PostGIS*, но для распределенного колоночного хранилища *Accumulo* [115], затем был передан в фонд *Apache* и работает с хранилищем *HBase*. *GeoJinni* — расширение для *Hadoop*, добавляющее геопространственные функции в различные слои и компоненты *Hadoop* для хранения, обработки и индексации больших массивов геоданных. Хранятся и обрабатываются геоданные как ключ-значение. Добавляются инструменты для загрузки и выгрузки различных форматов геоданных. *GeoJinni* создает два слоя индексного пространства, локальный и глобальный. Глобальный индекс позволяет партиционировать данные по *DataNode* кластера, локальный — отвечает за партиции на каждом *DataNode*. *GeoMesa* — это набор инструментов, созданный специально для распределенной обработки, анализа и визуализации больших пространственно-временных данных, в том числе и потоковых. Для хранения наборов массивов данных служат распределенные колоночные хранилища, такие как *Accumulo, HBase, Google Bigtable*. Также *GeoMesa* позволяет обрабатывать данные практически в реальном времени через специальный слой для системы потоковых сообщений *Apache Kafka* [115].

Pivotal GemFire — in-memory data grids (*IMDG*)[20], технология распределенных хранилищ данных в памяти, работающая на *Apache Geode*, являются ключом к работе современных высокоскоростных приложений с интенсивной обработкой данных.

Pivotal GPText, основанный на *Apache SolrCloud*, осуществляет поиск в корпоративной среде (социальные сети, электронная почта) свободного текста и, используя библиотеку *Apache MADlib*, обеспечивает масштабную аналитическую обработку для поддержки принятия бизнес-решений.

И в заключение хочу сослаться на высокие оценки аналитиков *Gartner* [67] о решениях *Pivotal*, в частности, подчеркивается их пригодность для науки о данных (*Data Science*): «как представляется, особенно они подходят для контекстно-независимых случаев использования *Pivotal Greenplum* с ее аналитикой в БД (*in-database analytics*) совместно с *Apache MADlib*».

Общие выводы

Проанализировав сравнительно небольшой ряд аналитических СУБД для Больших Данных (*BigDate*) широко известных производителей в мировом ИТ-сообществе сквозь призму трансформации существующих методов обработки и инфраструктуры в решения, определяемые *IDC* как новое поколение технологий и архитектур, предназначенных для извлечения экономичной выгоды из очень больших объемов разнообразных данных, обеспечивающих высокую скорость съема и анализа, можно заключить следующее.

Наряду с уже общепризнанной архитектурой массового параллелизма обработки (*Massively Parallel Processing, MPP*) в классе вычислительных систем, состоящих из множества узлов, организованных по принципу *shared nothing*, все чаще применяют кластер *Apache Hadoop* – широко известную программную инфраструктуру, в которую интегрированы ряд модулей, программных каркасов (*frameworks*) с теми или иными целевыми функциями, позволяющими создать полнофункциональную платформу хранения и обработки неструктурированных данных.

Возникшая дилемма: возможность обработки в рамках одного хранилища разнородной информации высокопараллельными, основанными на языке *SQL* системами, обеспечивающими полное соблюдение *ACID*, и распределенными системами *Hadoop*, быстро ставшим предпочтительным при работе с не-

[20] В данном случае *Grids* трактуется как объединение в единую распределенную инфраструктуру ресурсов памяти серверов кластера [100].

структурированной информацией, имеет ряд интересных решений интеграции, но два из них заслуживают особого внимания. Одно из них — проект *HAWQ* — это уровень реляционной базы данных, расположенный поверх распределенной файловой системы *Hadoop* (*HDFS*). *HAWQ* записывает и считывает данные из *HDFS* изначально. *HAWQ* — это собственный механизм запросов *SQL Hadoop*, который сочетает в себе ключевые технологические преимущества базы данных *MPP* с масштабируемостью и удобством *Hadoop*. Другое — хранилище, представляемое унифицированной архитектурой данных *Teradata Unified Data Architecture*™, и платформа управления данными, учитывающая все варианты применения хранилищ: традиционное, операционное, логическое и контекстно-независимое. Высокопроизводительный доступ к данным, обработку и виртуальную доставку к системам в гетерогенных аналитических средах обеспечивает экосистема *Teradata QueryGrid*™ — своеобразная матрица, использующая параллельное перемещение данных между объектами обмена. Идея экосистемного подхода для охвата разного типа данных, сводится к связыванию узловых информационных точек в разных средах. Принятая унифицированная архитектура данных *Teradata*® *UDA*™ не противоречит единому представлению данных, без их перемещения, — концепту логических хранилищ данных и подчеркивают статус *LDW* как окончательного решения для БД и для аналитики.

Аналитические возможности исследованных систем не ограничиваются только *SQL*-анализом. Так, *Greenplum* расширяет возможности *SQL* через многоязычные пользовательские функции в таких языках как *Python, R, Java, Perl, C/C ++*. *Teradata* предоставляет готовые к использованию выразительные функции *SQL-MapReduce*® и *Graph* для высокопроизводительной аналитики, функции временных рядов, аналитики текста и многое другое для исследования *BigDate*. Аналитические механизмы (*SQL, SQL-MapReduce* и *SQL-Graph*) обеспечивают оптимальную обработку аналитических задач в больших объемах данных, например, полная обработка сетевой аналитики (*SNA*P) в *Teradata* позволяет вызвать одним *SQL*-запросом расширенную аналитику нескольких видов (граф, путь/шаблон, текст, *SQL* и статистический прогнозный анализ).

Наличие в разработках библиотек масштабируемых алгоритмов машинного обучения *Machine Learning* предоставляет *in-database analytics* и обеспечивает возможность интеллектуального анализа данных при более объяснимых моделях, что определенно приближает нас при наличии развитых механизмов выявления «нужных данных» и библиотек графического отображения информации, к платформам для научных исследований данных.

Ежегодный рост *BigDate* обуславливает повышение скорости обработки данных. Использование аналитических вычислений в оперативной памяти БД (*in-database analytics*), исключающее перемещение данных и сокращающее время обработки стало уже привычным.

Более широкое использование памяти (*in-memory computing, IMC*) — платформы, работающие в компьютерной памяти, еще сравнительно редки. Так, в *Kx Systems* (*kdb+*), благодаря *IMC* была реализована архитектура гибридной транзакционно/аналитической обработки — *HTAP* (*hybrid transaction/analytical processing*), позволяющая приложениям анализировать данные по мере их поступления и обновления функциями обработки транзакций.

Расширенная аналитика в режиме реального времени, такая как прогнозирование и моделирование, стала неотъемлемой частью наблюдаемого процесса, а не позиционируется как отдельное действие, выполненное после. *Teradata* реализовала инновационную технологию баз данных *Intelligent Memory Teradata* (*IMDBMS* от *Teradata*) — расширенное пространство памяти за пределами кеша, что значительно увеличило производительность запросов и обеспечило эффективную технологию хранения разнообразных данных в памяти. *Pivotal GemFire* разработала технологию *in-memory data grids* (*IMDG*) распределенных высокопроизводительных хранилищ данных

в памяти для работы современных высокоскоростных приложений с интенсивной обработкой данных.

Технологии *IMC*, такие как система БД в оперативной памяти (*IMDBMS*) и высокомасштабируемые, отказоустойчивые хранилища данных в оперативной памяти *in-memory data grids* низкой латентности будут востребованы, в том числе и для расширенной аналитики, такой как прогнозирование и моделирование. *Hadoop* и компоненты экосистемы его продуктов достаточно широко представлены и функциональны, чтобы удовлетворить требованиям обработки *BigDate*.

Технология в целом отработана, ее признали известные в мире ИТ-компании и активно использовали для разработки инфраструктурных решений прогрессивных средств, в том числе и расширенной аналитики для бизнес-анализа и платформ научных исследований данных.

Наступившая фаза относительной стабильности (*commodity phase*) говорит о том, что технология становится обычной и доступной для всех.

REFERENCES

90. *Greenplum Database*, [online] Available at:<https://greenplum.org/>[Accessed 11 Jun. 2018].
91. *Greenplum® Database* 4.1 Administrator Guide, [online] Available at:<media.gpadmin.me/wp-content/uploads/2011/05/GP-4100-AdminGuide.pdf >[Accessed 11 Jun. 2018].
92. *EMC Greenplum Data* Computing Appliance: Performance and Capacity for Data Warehousing and Business Intelligen¬ce, [online] Available at: Aug. 2011. https://japan.emc.com/microsites/japan/techcommunity/pdf/h8778-Greenplum DCA-HighCapacity-wp.pdf. [Accessed 11 Jun. 2018].
93. *Load and Go*: Fast Data Loading with the Greenplum Data Computing Appliance (DCA) // Massive data news – https://www.emc.com/collateral/hardware/white-papers/load-and-go-fast-data-loading-greenplum-data-computing-appliance-wp.pdf >[Accessed 7 Aug. 2018].
94. *EMC Greenplum Data* Computing Appliance. Driving the future of data warehousing, [online] Available at:<https://www.ens-inc.com/FileLibrary/2f9a80b2-a267-4c72-a9d6-3952dae13894/> [Accessed 11 Jun. 2018].
95. *Hill D. Greenplum*: EMC's Latest Plum? 10/25/2010. https://www.networkcomputing.com/storage/greenplum-emcs-latest-plum/1870771227>[Accessed 7 Aug. 2018].
96. *Pivotal Greenplum*®. Greenplum Database Concepts, [online] Available at: https://gpdb.docs.pivotal.io/580/admin_guide/intro/partI.html >[Accessed 7 Aug. 2018].
97. *Pivotal*. The World's First Open-Source Based, Multi-Cloud Data Platform Built for Advanced Analytics, [online] Available at:<-https://pivotal.io/pivotal-greenplum >[Accessed 7 Aug. 2018].
98. *Oursatyev A.A.*, 2018. "Big Data. Analytical Databases and Warehouse: Teradata". Upravlausie sistemy i masiny, 2, pp. 51 — 67. (In Russian).
99. *Oursatyev A.A.*, 2019. "Big Data. Analytical Databases and Warehouse: NETEZZA". Upravlausie sistemy i masiny, 1, pp. 51 — 67. (In Russian).
100. *Oursatyev A.A.*, 2018. "Big Data. Analytical Databases and Warehouse: Vertica, Kdb". Upravlausie sistemy i masiny, 1, pp. 57 —70. (In Russian).
101. *New Data Loading Technology* from Greenplum Offers Breakthrough Speeds For Large-Scale Data Warehousing. – San Mateo, CA (PRWEB), 2009 – http://www.prweb.com/releases/2009/03/prweb2235864.htm
102. *Pivotal Greenplum*®. About Parallel Data Loading. https://gpdb.docs.pivotal.io/580/admin_guide/intro/ about_loading.html.
103. *Pivotal Greenplum*®. Choosing the Table Storage Model. https://gpdb.docs.pivotal.io/580/admin_guide/ddl/ddl-storage.html.
104. *Storage Comes At a Price,* 2016. https://www.linkedin.com/pulse/storage-comes-price-sandeep-katta-
105. *Pivotal Greenplum*®. About Concurrency Control in Greenplum Database. https://gpdb.docs.pivotal.io/580/admin_guide/ in-tro/about_mvcc.html
106. *Pivotal Greenplum* v5.1. About GPORCA. https://gpdb.docs.pivotal.io/510/admin_guide/query/topics/query-piv-optimizer.html.

107. *Пакет Pivotal Big* Data Suite ускоряет цифровую трансформацию, 2015. http://www.storagenews.ru/news_take.asp?Code=2319.
108. *Graefe G. Volcano* — An Extensible and Parallel Query Evaluation System, 1994. https://www.researchgate.net/publication/3296396_Volcano-An_Extensible_and_Parallel_ Query_Evaluation_System
109. *Graefe G.* The Cascades Framework for Query Optimization, 1995. https://www.researchgate.net/publication/220282640_The_Cascades_Framework_for_Query_ Optimization.
110. *A Modular Query Optimizer* Architecture for Big Data // Mohamed A. Soliman, Lyublena Antova, Venkatesh Raghavan and etl. https://content.pivotal.io/white-papers/orca-a-modular-query-optimizer-architecture-for-big-data.
111. *Addison Huddy*. GPORCA, A Modular Query Optimizer, Is Now Open-Source. Pivotal Engineering Journal, 2016. http://engineering.pivotal.io/post/gporca-open-source/.
112. *Data Lake* – универсальное хранилище для аналитики больших данных. http://www.storagenews.ru/60/EMC_Data_Lake_60.pdf.
113. *Серов Д.* Машины для аналитиков. «ОС», 2011. N 04., 2011. https://www.osp.ru/os/2011/04/13008766/
114. *New Features* in Greenplum Database 3.2. Welcome to Greenplum Database 3.2.0.0. http://docs.huihoo.com/greenplum/ GPDB-3.2.0.0-README.pdf
115. *Oursatyev A.A.*, 2016. “Some Big Data Analytics Software Environments”. Upravlausie sistemy i masiny, 3, pp. 29 — 42. (In Russian).
116. *Lozinskiy A.P., Simakhin V.M., Oursatyev A.A.*, 2017. “Technologies modeling for processing large data on the local cloud plat-form”. Upravlausie sistemy i masiny, 03, pp. 6–19. (In Russian).
117. *Greenplum Database* 4.3.6.1 Release Notes, 2015. http://docs.huihoo.com/greenplum/pivotal/4.3.6/relnotes/GPDB_4361_README.html#topic36
118. *Bodkin Ron*. MapR Releases Commercial Distributions based on Hadoop. InfoQ, 2011. https://www.infoq.com/news/2011/07/mapr.
119. *Harris Derrick*. Startup MapR Underpins EMC’s Hadoop Effort, 2011. https://gigaom.com/2011/05/25/startup-mapr-underpins-emcs-hadoop-effort/.
120. *Clark Jack*. EMC taps MapR technology for Hadoop distro, 2011. https://www.zdnet.com/article/emc-taps-mapr-technology-for-hadoop-distro/.
121. *Chapter 7*: Loading and Unloading Data. Greenplum® Database 4.2 Database Administrator Guide. Rev: A07, 2014 (4.2.7.1). https://www.emc.com/collateral/TechnicalDocument/docu44316.pdf.
122. *Chapter 7*: Loading and Unloading Data. Greenplum® Database Version 4.3 Database Administrator Guide. Rev: A01. © 2013. GoPivotal, Inc. https://gpdb.docs.pivotal.io/4300/pdf/ GPDB43_DBAGuide.pdf.
123. *External Table* Support for Avro and Parquet File Formats on HDFS. Greenplum Database 4.3.6.1 Release Notes. Sept. 2015. http://docs.huihoo.com/greenplum/pivotal/4.3.6/relnotes/GPDB_4361_README.html#topic36.
124. *Jeffrey Wang*. Dremel. Data Model. Ternary Search, 2013. http://ternarysearch.blogspot.com/2013/06/dremel-data-model.html.
125. *Dremel*: Interactive Analysis of Web-Scale Datasets / Melnik Sergey, Andrey Gubarev, Jing Jing Long et. al., Int. Conf. on Very Large Data Bases, 13–17 Sept. 2010, Singapore. https://static.googleusercontent.com/media/research.google.com/ ru//pubs/archive/36632.pdf.
126. *Diwakar Kasibhotla*. Greenplum and Hadoop HDFS integration, 2012. https://dwarehouse.wordpress.com/2012/10/10/greenplum-and-hadoop-hdfs-integration/.
127. *New Functionality* in Greenplum Database 4.2. Welcome to Greenplum Database 4.2. Updated: Nov. 23. 2011. http://media.gpadmin.me/wp-content/uploads/2012/11/ GPDB_4200_README.pdf.
128. *Harris Derrick*. EMC Makes a Big Bet on Hadoop, 2011. https://gigaom.com/2011/05/09/emc-hadoop/
129. *Greenplum Platform* Extension Framework (PXF). Using PXF with External Data. Pivotal Greenplum v5.5.0 Documentation. https://gpdb.docs.pivotal.io/550/pxf/overview_pxf.html.
130. *Using PXF* to Read and Write External Data. Greenplum Platform Extension Framework (PXF). https://gpdb.docs.pivotal. io/550/pxf/using_pxf.html
131. *Greenplum 5*: первые шаги в Open Source. Блог компании IBS, 2017. https://habr.com/company/ibs/blog/343640/
132. *Menninger David*. EMC Enters Elephant Race with Hadoop, 2011. https://davidmenninger.ventanaresearch.com/2011/05/12/emc-enters-elephant-race-with-hadoop
133. *MapR Technologies* and EMC Announce Technology Licensing Agreement for Next Generation Hadoop Distribution, 2011. https://mapr.com/company/press/mapr-technologies-and-emc-announce-technology-licensing-agreement-next-generation/

134. *MapR File System* (MapR-FS). https://mapr.com/docs/52/MapROverview/c_maprfs.html
135. *Direct Access* NFS. https://mapr.com/docs/52/MapROverview/c_direct_NFS.html
136. *MapR System* Overview. https://mapr.com/docs/52/MapROverview/ c_overview_intro.html
137. *MapReduce Version* 1. https://mapr.com/docs/52/MapROverview/c_mrv1.html?hl =directshuffle
138. *MapR Technologies*: How Direct Shuffle actually works? https://www.quora.com/MapR-Technologies-How-Direct-Shuffle-actually-works
139. *White T. Hadoop*: The Definitive Guide. 1st ed. Sebastopol: O'Reilly Media, 2009. 528 p. http://oreilly.com/catalog/9780596521981
140. *Hadoop Acceleration* Through Network Levitated Merge. Wang Y., Que X., Yu W. https://www.cs.fsu.edu/~yuw/pubs/2011-SC-Yu.pdf
141. *JVM-Bypass* for Efficient Hadoop Shuffling. Wang, Xu C., Li X., Yu W., IPDPS '13 Proceedings of the 2013 IEEE 27th Interna-tional Symposium on Parallel and Distributed Processing, pp. 569–578. https://www.cs.fsu.edu/~yuw/pubs/2013-IPDPS-Yu.pdf
142. *S. Rao. I-files:* Handling Intermediate Data In Parallel Dataflow Graphs (Sailfish). https://www.cics.umass.edu/event/i-files-handling-intermediate-data-parallel-dataflow-graphs
143. Camdoop: exploiting in-network aggregation for big data applications / P. Costa, A. Donnelly, A. Rowstron et al., In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, NSDI'12, pages 3–3, Berkeley, CA, USA, 2012. USENIX Association. http://www.cs.yale.edu/homes/yu-minlan/teach/csci599-fall12/papers/nsdi12-final11_0.pdf
144. *Direct Shuffle* on YARN. https://mapr.com/docs/home/MapROverview/c_direct_shuffle_ yarn.html
145. *EMC Greenplum* HD Enterprise Edition. Advancing Hadoop for the Enterprise – Copyright © 2011 EMC Corporation. Published in the USA. 08/11 Data Sheet H8892. http://www.netdyninc.com/sw/swchannel/images/ProductCatalog/Product-Page/File/datasheet69.pdf
146. *MC Greenplum*® HD Enterprise Edition. Administrator Guide Rev: A01, 2011. EMC Corporation. https://www.emc.com/ collat-eral/TechnicalDocument/docu34982.pdf
147. *Aslett Matthew*. What's in a name? EMC Greenplum rebrands its Hadoop distros, 2012. https://blogs. the451group.com/information_management/2012/01/31/whats-in-a-name-emc-greenplum
148. *Горизонтально масштабируемая* сетевая система хранения данных для Greenplum HD. Система для хранения и анализа больших данных EMC Isilon. https://ukraine.emc.com/collateral/hardware/solution-overview/h8319-scale-out-nas-greenplum-hd-so.pdf-re-brands-its-hadoop-distros/
149. *DELL EMC* Isilon Big Data Storage and Analytics Solutions. Efficient, Flexible In-Place Hadoop Analytics. https://www.emc.com/collateral/hardware/solution-overview/h8319-scale-out-nas-greenplum-hd-so.pdf
150. *Обзор работы* HDFS с OneFS. http://doc.isilon.com/onefs/hdfs/02-ifs-c-hdfs-conceptual-topics.htm
151. *Patel Mona*. Chorus Brings Data Science Minds Together, 2013. https://blog.dellemc.com/en-us/chorus_data_science/
152. *The Age of Self-Service* Data is Upon Us. https://go.unifisoftware.com/Definitive-Guide-to-Self-Service-Data
153. *Greenplum Software* Introduces Greenplum Chorus, 2010. http://www.b-eye-network.com/ view/13182
154. *Howard Philip*. Self-service data preparation and cataloguing, 2016. https://www.bloorresearch.com/research/self-service-data-preparation-cataloguing-p2/
155. *Apache HAWQ* is Apache Hadoop Native SQL. Advanced Analytics MPP Database for Enterprises. http://hawq.apache.org/
156. *Prickett Morgan* Timothy. EMC morphs Hadoop elephant into SQL database HAWQ, 2013. https://www.theregister.co.uk/2013/02/25/emc_pivotal_hd _hadoop_hawq_database/
157. *Kersteter Bart*. What is HAWQ?, 2013. https://www.quora.com/What-is-HAWQ
158. *Pivotal HDB* 2.1.1 Documentation, 2017. https://hdb.docs.pivotal.io/211/hawq/overview/ HAWQOver-view.html
159. *Pivotal Greenplum*: Open-Source, Massively Parallel Data Platform for Advanced Analytics. https://content.pivotal.io/datasheets/pivotal-greenplum

Received 03.04.2019

A.A. Oursatyev, PhD in Techn. Sciences, Leading Research Associate, Senior Scientists,
International Research and Training Centre of Information Technologies and Systems
of the NAS and MES of Ukraine, Glushkov ave., 40, Kyiv, 03187, Ukraine,
aleksei@irtc.org.ua

BIG DATA. ANALYTICAL DATABASES AND DATA WAREHOUSE: GREENPLUM

Introduction. The article is a continuation of the *Big Data* and tools study, which is transformed into technology of the new generation and architecture of the *BD* platforms and storage for the intelligent output. In this part the review of *DB Netezza* is presented. The main attention is paid to the issues of changing the infrastructure, the tool environment and the platform for identifying the necessary information and new knowledge from the *Big Data*, the initial information about the product is given in the product general description.

Purpose. The purpose is to consider and evaluate the application effectiveness of the infrastructure solutions for new developments in the *Big Data* study, to identify new knowledge, the implicit connections and indepth understanding, insight into phenomena and processes.

Methods. The informational and analytical methods and technologies for data processing, the methods for data assessment and forecasting, taking into account the development of the most important areas of the informatics and information technology.

Results. *Greenplum*, as well as *Netezza* and *Teradat*a, created its *Data Computing Appliance* (*DCA*) complex, and later, an analytical *Pivotal database Greenplum Database* of corporate class with powerful and fast analytics for large data volumes under the *Pivotal* trademark. The relational database uses *Postgres Core's* largescale parallel *Shared Nothing MPP* architec-ture. The internal elements of *PostgreSQL* have been modified or added to support the parallel structure of the *Greenplum* database. Introduced technology *Greenplum MPP Scatter/Gather Streaming* fast loading (*unloading*), polymorphic data storage. For mass loading and reading of data, the *Append*-optimized or append-only storage format is used, providing performance advantages over *Heap* tables. The concurrency control in the *Greenplum PostgreSQL* database occurs without using a lock to control concurrency. Data consistency is supported by *Multiversion Con*-currency *Control, MVCC*, which provides isolated transaction for each database session. *GPORCA* is *ORCA* query optimizer which expands the capabilities of planning and optimizing the legacy of *GPQUERY*.

For the first time, *EMC* announced its *Big Data* analytics development programme in 2011. A comprehensive strategy for the integration and support of open source software *Apache Hadoop* was presented. As a result of merging the *Greenplum* database with *Hadoop*, it became possible to expand data types in analytical studies within a single repository. The adaptation of the *Greenplum* database to the *HDFS* distributed file system has been completed for more than two years by the largest development team of *Hadoop*. It was necessary to create a storage layer that would improve the current *HDFS* version in terms of performance, availability, and ease of use. Along with *Cloudera, Hortonworks* and *MapR*, the largest providers of data storage and processing solutions, *EMC, Greenplum, Pivotal* appeared on the market of services providing infrastructure support for *Hadoop*, promoting their own *Hadoop* distribution.

We have considered the technologies being used to modify the Hadoop infrastructure for the analytics platform in *Greenplum* products and the *RDBMS* integration environment with *Hadoop*: from the *Writable External Table* to the expansion of the *Greenplum* database (*PXF*), which allows exchanging data with third-party heterogeneous systems, *Pivotal Greenplum HAWQ* - self-contained *SQL Hadoop* query engine, which combines the key technological advantages of the *MPP* database with the scalability and convenience of *Hadoop*, and the *Greenplum Chorus* self-service data platform.

General conclusions. Having analysed a relatively small number of analytical *DBMS* for *BigDate* by well-known manufacturers in the global *IT* community through the transforming existing processing methods and infrastructure into solutions, identified by *IDC* as a new genera-tion of technologies and architectures designed to extract economical benefits from a very large amounts of diverse data providing the high rate of removal and analysis, we can conclude the following.

1. Along with the generally recognized *MPP* architecture, the *Apache Hadoop* cluster is a well-known software infrastructure that integrates a number of modules (frameworks) with the various target functions, a class of computing systems consisting of many nodes organized on the basis of shared nothing — the ability to create a full-featured platform for storing and processing unstructured data.

The arisen dilemma is that the ability to process different types of data within a single reposi-tory with highly parallel, *SQL*-based systems that ensure full compliance with *ACID* and distributed *Hadoop* systems, which have quickly become the preferred way of working with unstructured information, has a number of interesting integration solutions, but two of them deserve special attention. One of them, the *HAWQ* project, is a relational database tier located on top of the *Hadoop* distributed file system's *HDFS*. *HAWQ* writes and reads data initially from *HDFS*. *HAWQ* is a proprietary *SQL Hadoop* query

engine that combines the key technological benefits of the *MPP* database with the scalability and convenience of Hadoop. The other is the storage, represented by the *Teradata Unified Data Architecture*, and the data management platform that takes into account all the storage applications: traditional, operational, logical, and context-independent. High-performance data access, processing and virtual delivery to systems in heterogeneous analytic environments is provided by the *Teradata QueryGrid* ™ ecosystem, a kind of matrix that uses parallel data movement between exchange objects. The idea of an ecosystem approach to cover different types of data comes down to linking nodal information points in the different environments. The adopted of *Teradata® UDA* ™ does not contradict the unified presentation of data, without moving them, to the concept of logical data stores and emphasize the status of *LDW* as a final solution for the data-base and for analytics.

2. Analytical capabilities of the studied systems are not limited to *SQL*-analysis. So, *Greenplum* extends the capabilities of SQL through multilingual user functions in languages such as *Python, R, Java, Perl, C/C++*. *Teradata* provides ready-to use *SQL-MapReduce®* and *Graph* expressive functions for high-performance analytics, time series functions, text analytics, and more for *BigDate* research. Analytical mechanisms (*SQL, SQL-MapReduce* and *SQL-Graph*) ensure an optimal processing of the analytical tasks in large amounts of data, for example, full network analytics processing (*SNAP*) in *Teradata* allows to call several types of advanced analytics (graph, path/template, text, *SQL* and statistical forecast analysis).

3. In the libraries development there are the scalable algorithms *Machine Learning* and the use of cluster computing in *RAM* will ensure the possibility of AI with more explicable models, which will definitely bring us closer to *Data Science* platforms, if there are developed mechanisms for identifying the "*needful data*" and libraries of graphic information display.

4. The annual growth of *BigDate* causes an increase in data processing speed. The use of in-database analytical computing, which eliminates data movement and reduces processing time, has become customary. *Wider* use (in-memory computing, *IMC*) — platforms operating in computer memory are still relatively rare. For example, *Kx Systems* (*kdb +*), thanks to *IMC*, implemented the hybrid transaction/analytical processing, *HTAP* architecture, allowing applications to analyse data as it arrives and is updated with transaction processing functions. Extended real-time analytics, such as forecasting and modelling, have become an integral part of the observed process, rather than being positioned as a separate action performed after. *Teradata* implemented an innovative database technology *Intelligent Memory Teradata* (*IMDBMS from Teradata*) — expanded memory space outside the cache, which significantly increased epy query performance and provided an effective technology for storing a variety of data in memory. *Pivotal GemFire* has developed in-memory data grids (*IMDG*) technology for distributed high-performance in-memory data storage for modern high-speed applications with intensive data processing.

IMC technologies such as in-memory database (*IMDS*) and high-scale, fault-tolerant in-memory data storage grids of low latency will be in demand, especially for advanced analytics such as prediction and modelling.

5. *Hadoop* and the components of the ecosystem of its products are fairly well represented and functional to meet the requirements of *BigData* processing. The technology as a whole has been developed, it has been recognized by well-known *IT* companies in the world and are actively used for developing infrastructure solutions of the advanced tools, including advanced analytics for business analysis and *Data Science* platforms. The coming phase of relative stability (*commodity phase*) indicates that the technology is becoming common and accessible to all.

Keywords: *Greenplum Data Computing Appliance (DCA), Non Shared Nothing MPP architecture based on PostgreSQL core, a MPP Scatter/Gather Streaming technology the data loading and unloading, Polymorphic Data Storage, BigData analytics, integration platforms, Own SQL query engine for Hadoop HAWQ, self-service data.*

О.А. Урсатьєв, канд. техн. наук, старший науковий співробітник,
провідний науковий співробітник, Міжнародний науково-навчальний центр
інформаційних технологій та систем НАН та МОН України,
просп. Глушкова, 40, Київ 03187, Україна,
aleksei@irtc.org.ua

ВЕЛИКІ ДАНІ. АНАЛІТИЧНІ БАЗИ ДАНИХ І СХОВИЩА: GREENPLUM

Вступ. Стаття є продовженням досліджень Великих Даних і інструментарію, що трансформується в нове покоління технологій і архітектури платформ БД та сховищ для інтелектуального виводу. У даній частині огляду подано *DB Greenplum*. Основну увагу приділено питанням зміни інфраструктури, інструментального середовища і платформи для виявлення необхідної інформації та нових знань з Великих Даних, а початкові відомості про продукт наведено в загальній характеристиці виробу.

Мета. Розглянути та оцінити ефективність застосування інфраструктурних рішень нових розробок в дослідженнях Великих Даних для виявлення нових знань, неявних зв'язків і поглибленого розуміння, проникнення в суть явищ і процесів.

Методи. Інформаційно-аналітичні методи і технології обробки даних, методи оцінки та прогнозування даних, з урахуванням розвитку найважливіших галузей інформатики та інформаційних технологій.

Результати. *Greenplum*, так само як *Netezza* і *Teradata*, створив свій комплекс *Data Computing Appliance*, пізніше – аналітичну БД *Pivotal Greenplum Database* корпоративного класу з потужною і швидкою аналітикою для великих обсягів даних під торговою маркою *Pivotal*. Реляційна БД використовує широкомасштабну паралельну *MPP*-архітектуру без розподілу ресурсів на основі ядра *Postgres Core*. Внутрішні елементи *PostgreSQL* були модифіковані або доповнені для підтримки паралельної структури БД *Greenplum*. Впроваджено технологію *Greenplum MPP Scatter/Gather Streaming* швидкого завантаження (вивантаження), поліморфного зберігання даних. Для масового завантаження і читання даних використовується *Append*-оптимізований формат зберігання, що забезпечує переваги продуктивності в порівнянні з таблицями *Heap*. Управління паралелізмом в БД *Greenplum* і *PostgreSQL* відбувається без використання блокування для його контролю. Узгодженість даних підтримується мультиверсійною моделлю контролю конкурентних транзакцій *MVCC*, що забезпечує ізоляцію транзакцій для кожного сеансу БД. *GPORCA* – оптимізатор запитів *ORCA* розширює можливості планування і оптимізації успадкованого *GPQUERY*.

Вперше *EMC* анонсувала свою програму розвитку аналітики *Big Data* в 2011 р. Була представлена комплексна стратегія інтеграції та підтримки ПЗ з відкритим вихідним кодом *Apache Hadoop*. В результаті злиття БД *Greenplum* з *Hadoop* з'явилася можливість в рамках одного сховища розширити типи даних в аналітичних дослідженнях. Адаптація БД *Greenplum* для розподіленої файлової системи *HDFS* виконувалася протягом більше двох років найбільшої командою розробників *Hadoop*, так як було потрібно створити рівень зберігання, який зміг би поліпшити поточну версію *HDFS* з точки зору продуктивності, доступності та простоти використання. На ринку послуг, що надають підтримку інфраструктури *Hadoop*, поряд з компаніями *Cloudera, Hortonworks* і *MapR*, з'явилися постачальники рішень для зберігання і обробки даних в особі *EMC*, *Greenplum* і *Pivotal*, які також стали просувати свій власний дистрибутив *Hadoop*.

Розглянуто використовувані технології, що модифікують інфраструктуру *Hadoop* для платформи аналітики в продуктах *Greenplum*, і середовища інтеграції реляційної БД з *Hadoop*: від записуваних зовнішніх таблиць до розширення БД *Greenplum* (*PXF*), що дозволяє обмінюватися даними зі сторонніми гетерогенними системами; *Pivotal Greenplum HAWQ* – власного механізму запитів *SQL Hadoop*, що поєднує в собі ключові технологічні переваги бази даних *MPP* з масштабованістю і зручністю *Hadoop*; платформу *self-service data Greenplum Chorus*.

Загальні висновки. Проаналізувавши порівняно невеликий ряд аналітичних СУБД для Великих Даних широко відомих виробників в світовому IT-співтоваристві крізь призму трансформації існуючих методів обробки та інфраструктури в рішення, які визначаються *IDC* як нове покоління технологій і архітектур, призначених для вилучення економічної вигоди з дуже великих обсягів різного типу даних, що забезпечують високу швидкість знімання і аналізу, можна зробити наступний висновок.

1. Поряд з вже загальновизнаною архітектурою масового паралелізму обробки, *MPP* в класі обчислювальних систем, що складаються з безлічі вузлів, організованих за принципом *shared nothing*, застосовують кластер *Apache Hadoop* з широко відомою програмною інфраструктурою, в яку інтегровані ряд модулів, програмних каркасів (*frameworks*), з тими чи іншими цільовими функціями, що дозволяють створити повнофункціональну платформу зберігання і обробки неструктурованих даних.

Виниклу дилему – можливість обробки в рамках одного сховища різнорідних типів даних високопаралельними, заснованими на мові *SQL* системами, що забезпечують повне дотримання *ACID*, і розподіленими системами *Hadoop*, які швидко стали кращими при роботі з неструктурованою інформацією, мають ряд цікавих рішень інтеграції, але два з них заслуговують на особливу увагу. Одне з них – проект *HAWQ* – це рівень реляційної бази даних, розташований поверх розподіленої файлової системи *Hadoop* (*HDFS*). *HAWQ* записує і зчитує дані з *HDFS* спочатку. *HAWQ* – це власний механізм запитів *SQL Hadoop*, який поєднує в собі ключові технологічні переваги бази даних *MPP* з масштабованістю і зручністю *Hadoop*. Друге рішення – сховище, яке надається уніфікованою архітектурою даних *Teradata Unified Data Architecture*™, і платформа керування даними, що враховує всі варіанти застосування сховищ: традиційне, операційне, логічне і контекстно-незалежне. Високопродуктивний доступ до даних, обробку і віртуальну доставку до систем в гетерогенних аналітичних середовищах забезпечує екосистема *Teradata QueryGrid*™ – своєрідна матриця, яка використовує паралельне переміщення даних між об'єктами обміну. Ідея екосистемного підходу для охоплення різного типу даних, зводиться до зв'язування вузлових інформаційних точок в різних середовищах. Прийнята уніфікована архітектура даних *Teradata® UDA*™ не суперечить єдиному поданню даних, без їх переміщення, – концепту логічних сховищ даних і підкреслюють статус *LDW* як остаточного рішення для БД і аналітики.

2. Аналітичні можливості досліджених систем не обмежуються тільки *SQL*-аналізом. Так, *Greenplum* розширює можливості *SQL* через призначені для користувача функції в таких мовах як *Python, R, Java, Perl, C / C ++*. *Teradata* надає готові до використання виразні функції *SQL-MapReduce*® і *Graph* для високопродуктивної аналітики, функції часових рядів, аналітики тексту і багато іншого для дослідження *BigDate*. Аналітичні механізми (*SQL, SQL-MapReduce* і *SQL-Graph*) забезпечують оптимальну обробку аналітичних задач у великих обсягах даних, наприклад, повна обробка мережевої аналітики (*SNAP*) в *Teradata* дозволяє викликати одним *SQL*-запитом розширену аналітику декількох видів (граф, шлях/шаблон, текст, *SQL* і статистичний прогнозний аналіз).

3. Наявність в розробках бібліотек масштабованих алгоритмів машинного навчання *Machine Learning* надає аналітику, яка виконується в оперативній пам'яті і забезпечує можливість інтелектуального аналізу даних при більш з'ясовних моделях, що безумовно наближає нас при наявності розвинених механізмів виявлення «потрібних даних» та бібліотек графічного відображення інформації, до платформ наукових досліджень даних.

4. Щорічне зростання *BigDate* обумовлює підвищення швидкості обробки даних. Використання аналітичних обчислень в БД, що усуває накладні витрати при переміщенні великих наборів даних до аналітичних застосунків, стало вже звичним. Більш широке використання пам'яті (*in-memory computing*, *IMC*) – платформи, що працюють в комп'ютерній пам'яті, ще порівняно рідкісні. Так, в *Kx Systems* (*kdb +*), завдяки *IMC* була реалізована архітектура гібридної транзакційно/аналітичної обробки – *HTAP*, що дозволяє застосункам аналізувати дані по мірі їх надходження і поновлення функціями обробки транзакцій. Розширена аналітика в режимі реального часу, така як прогнозування і моделювання, стала невід'ємною частиною спостережуваного процесу, а не позиціонується як окрема дія, виконана після. *Teradata* реалізувала інноваційну технологію баз даних *Intelligent Memory Teradata* – розширений простір пам'яті за межами кеша, що значно збільшило продуктивність запитів і забезпечило ефективну технологію зберігання різноманітних даних в пам'яті. *Pivotal GemFir*e розробила технологію *in-memory data grids* (*IMDG*) розподілених високопродуктивних сховищ даних в пам'яті для роботи сучасних високошвидкісних за стосунків з інтенсивною обробкою даних.

Технології *IMC*, такі як система БД в оперативній пам'яті (*IMDS*) і високомасштабовані відмовостійкі сховища даних в оперативній пам'яті (*in-memory data grids*) низької латентності будуть затребувані, в тому числі і для розширеної аналітики – прогнозування і моделювання.

5. *Hadoop* і компоненти його екосистеми продуктів досить широко представлені і функціональні, щоб задовольнити майже всім вимогам *Big Data*. Технологія в цілому відпрацьована, її визнали відомі в світі IT-компанії і активно використовували для розробки і впровадження інфраструктурних рішень прогресивних засобів, в тому числі і розширеної аналітики для бізнес-аналізу і платформ наукових досліджень даних. Наступила фаза відносної стабільності (*commodity phase*), яка свідчить про те, що технологія стає звичайною і доступною для всіх.

Ключові слова: *Greenplum Data Computing Appliance, MPP-архітектура без розподілу ресурсів на основі ядра PostgreSQL, технологія MPP Scatter/Gather Streaming завантаження (вивантаження), поліморфне зберігання даних, аналітика Big Data, інтеграція платформ, механізм запитів SQL Hadoop HAWQ, самообслуговування даними.*