DOI https://doi.org/10.15407/usim.2018.01.057

УДК 004.65:004.7:004.75:004.738.5

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

БОЛЬШИЕ ДАННЫЕ. АНАЛИТИЧЕСКИЕ БАЗЫ ДАННЫХ И ХРАНИЛИЩА: *VERTICA, KDB*

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.*

Ключевые слова*: MPP — архитектура, HTAP — гибридная транзакционная/аналитическая обработка, LDW — логические хранилища данных, облачное хранение, платформа баз данных как услуга DBPaaS, аналитика по модели SaaS, среда управления данными, технология IMC.*

Введение

Появление Больших Данных (*Big Data*) — та отправная точка, с которой началась трансформация инструментальной среды аналитики в решения, направленные на поддержку согласованной обработки информации ряда источников путем предоставления сервиса данных, затребованных аналитическим приложением. Ключевой критерий — могут ли аналитические выводы, направленные на поддержку принятия решений, рассматриваться в контексте Больших Данных, — есть именно сочетание широкого спектра типов данных с учетом нетрадиционных форматов [1, 7, 13–15, 17]. Вместе с тем большие размеры наборов данных, их разнообразие и сложность вследствие нетрадиционных форматов, несопоставимость[1], короткий период актуальности некоторых из них и разные требования обработки, могут привести к невостребованности огромного количества накопленной информации [27].

Преодолеть это препятствие и превратить информацию в актив с наиболее существенными новыми показателями производительности и ценности информации — значит решить вопрос простого доступа к информации и возможность ее применения разными способами. Концепт логического *LDW*[2]-хранилища (*logical data warehouse*) позволит анализировать и управлять Большими Данными так, что-

[1] Несопоставимые данные — это данные, накопленные по нестандартным методикам.

[2] *LDW* — новая архитектура управления данными для аналитики, сочетающая сильные стороны традиционного хранилища с альтернативным управлением данными и стратегиями доступа, имеет семь основных компонентов: управление хранилищем; виртуализация данных; распределенные процессы; управление *SLA*; статистика аудита и услуги по оценке эффективности, разрешение таксономии и онтологии; управление метаданными [23, 24, 26, 27, 30].

бы единое предоставление разнообразных данных без их перемещения привело к тому, что содержащаяся в них информация стала явной, доступной и актуальной. Это один из новых перспективных подходов к хранилищам и управлению аналитическими данными [23–27, 30].

Этот подход согласуется с высказыванием *IDC*, определяющим методы обработки Больших Данных как новое поколение технологий и архитектур, предназначенных для экономичного извлечения выгоды из очень больших объемов различных данных, обеспечивающих высокую скорость съема, обнаружения и/или анализа [13, 14, 30].

Инфраструктурные решения Больших Данных

Развивая тему [30] о тенденции формирования хранилищ данных для аналитики и стратегии управления различными информационными ресурсами для предоставления эффективного доступа к ним, рассмотрим технологии, модели, методы и платформы аналитического вывода. О некоторых из них (массовые распределенные параллельные вычисления на кластерах, колоночная (столбчатая) модель хранения данных) уже упоминалось. Здесь уместно заметить, что узким местом аналитических систем с подавляющим большинством операций *чтение*, извлечение данных с дисков обычно — самая медленная часть запроса к базе данных (БД). Колоночные (*column-store*) БД, в сравнении с традиционными, позволяют на аналогичном оборудовании получить прирост скорости выполнения запросов от пяти–10 до 100 раз [28]. Вместе с тем они имеют недостатки: плохо работают при транзакционной нагрузке, медлительны в операциях запись — запись часто осуществляется крупными блоками (*bulk load*), — но очень хороши для аналитического доступа к большим наборам данных [29, 30].

Небезынтересно под этим углом зрения рассмотреть инфраструктурные решения с набором ключевых технологий, положенных в основу Больших Данных и используемых известными в мире ИТ-компаниями в новых разработках, ориентированных на задачи, в том числе расширенной, углубленной аналитики. Распространенные промышленные системы хранения и обработки Больших Данных и направление дальнейшего их развития дано в [31], а в [23, 24] — приведена оценка *Gartner* [3] ряда из них о занимаемом положении в мире среди подобных себе. В [31] также предпринята попытка классификации средств обработки в виде трех групп: Быстрые Данные (*Fast Data*), Большая Аналитика (*Big Analytics*) и Глубокое Проникновение (*Deep Insight*), т.е. способность глубоко проникать в суть, что позволило бы соотнести принятые технологии с ожидаемыми результатами их обработки.

Технологии работы с Большими Данными реализуются как на специализированных аппаратно-программных комплексах (*Greenplum Database*, *IBM Netezza*, *Vertica*, *Kdb*, *Teradata* и др.), так и в виде программных сред, как правило, с открытым исходным кодом (например, *frameworks Hadoop*, *Hadoop YARN*, *Mahout, MapReduce*, *Spark*, *Spark-SQL*, *Storm*, *Tez*), объединяемых в системы обработки данных и успешно работающие на стандартном серверном оборудовании. В настоящем обзоре будут рассмотрены некоторые специализированные аппаратно-программные системы хранения и обработки из перечня [31] и проанализированы изменения инфраструктуры, инструментальной среды и платформы для извлечения необходимой информации и новых знаний о Больших Данных [30]. Основное внимание уделено трансформации известной среды БД, СУБД или хранилищ данных для интеллектуального вывода, а начальные сведения о кон-

[3] *Gartner*, *Inc.* (*NYSE*: *IT*) — ведущая мировая исследовательская и консультационная компания. Предоставляет стратегические рекомендации и проверенные прогрессивные методы, способствующие клиентам преуспеть в решении важнейших приоритетов. *Gartner* имеет более 13 тыс. сотрудников, обслуживающих клиентов на 11 тыс. предприятий в 100 странах.

кретном продукте приведены в общей характеристике изделия.

Специализированные аппаратно-программные решения

Vertica. Общая характеристика. Колоночно-ориентированная СУБД предназначена для работы в горизонтально масштабируемой среде и основана на архитектуре с массовым параллелизмом обработки (*Massively Parallel Processing — MPP*), оптимизирована для записи [32]. Это позволяет при выполнении «тяжелых» аналитических запросов решать аналитические задачи в режиме, близком к реальному времени, работая с большими объемами структурированных данных. Обеспечивает практически линейный прирост производительности в случае масштабирования от терабайт до петабайт и более данных, и при этом не нуждается в специализированных аппаратных решениях с использованием стандартной промышленной платформы x86. Поколоночное хранение допускает возможность сильно компрессировать данные, так как в одной колонке таблицы данные, как правило, однотипные. В *Vertica* применяется оптимизированная функция хранения при запатентованной компрессии по столбцам, что дает возможность считывать с дисков не всю запись, а только нужные поля, участвующие в запросе. Для значительного ускорения выполнения таких запросов у *Vertica* дополнительно реализуются *проекции* (*projections*) — оптимизированная коллекция колонок таблицы, посредством которых можно описать дублирующие структуры хранения данных в виде нужных полей таблиц со своей сегментацией, сортировкой и, при необходимости, группировкой полей, сохраняемых в одном блоке. Это позволяет определить различные типовые аналитические запросы к данным, и, создав проекции[4], выполнять их достаточно быстро (*blazing-fast speed*) — от 50 до 1000 раз быстрее, в сравнении с традиционными построчными СУБД, покрывая весь спектр запросов любого уровня сложности [32–34]. Данные проекций синхронно обновляются вместе с данными таблиц. Недостаток такого подхода — в дополнительных затратах на запись данных и хранение их избыточного объема. Однако при достаточном количестве типовых запросов по большим объемам данных проекции оправдывают себя.

Одновременная загрузка из нескольких источников данных чревата ограничением на монопольную запись в таблицу, т.е. в один момент времени в таблицу может добавлять или изменять данные только одна сессия. *Vertica* использует *Write Optimized Store* (*WOS*) — резидентную структуру данных для их краткосрочного хранения, размещаемую в специальной области оперативной памяти, которая позволяет выполнять загрузку сессиям с подтверждением транзакции без ожидания окончания работ по распределению и переносу данных на диск [34]. Помимо скорости вставки данных, улучшается и качество их хранения в базе — по мере заполнения новыми данными от сессий, *WOS* их собирает, сегментирует, сортирует и записывает в базу данных, снижая их дефрагментацию, неизбежную при большом количестве вставок данных множеством сессий [33, 34].

Новое в технологиях обработки. Для работы с накопленными данными *Vertica* снабжена стандартным *SQL*-интерфейсом (*ANSI SQL*-99), имеющим расширения для работы с аналитическими запросами. С версии 7.0 *HP Vertica Analytics Platform* интегрировала в свою платформу экосистему продуктов[5], состоящую из *Apache Hadoop* и ряда дополнительных модулей, что позволило ей создать специальную область хранения и обработки неструктурированных данных — *Flex Zone*, основанную на технологиях гибких (*flex or flexible*) таблиц

[4] В общем случае на одну таблицу может быть несколько проекций.

[5] Такие компании как *Cloudera*, *Hortonworks* и *MapR* предоставляют коммерческие услуги по поддержке инфраструктуры *Hadoop*.

[35]. Данные могут иметь некоторую структуру (например, *JSON*[6] и файлы данных с разделителями, например, файл с расширением *CSV*[7]), быть полуструктурированными или строго структурированными, но методом, о котором пользователю либо не известно, либо у него нет соответствующего инструментария. Термин *гибкие таблицы* используется для охвата данных такого рода.

Flex Tables[8] и связанные с ними функции хранения и управления неструктурированными данными используются для выполнения следующих задач: создание гибких таблиц, загрузка в них данных, применение стандартных *SQL*-запросов для извлечения данных из гибких таблиц. Таблицы *Alter flex* служат для добавления регулярных (материализованных) столбцов для интересующих данных и создания гибридной таблицы, содержащей как неструктурированные, так и структурированные данные.

Использование гибридных гибких таблиц приводит к аналогичной производительности для обычных колоночных таблиц *HP Vertica*. Однако обработка внешней таблицы происходит значительно медленнее, чем внутренней *HP Vertica*. Поэтому внешние таблицы используются только для нерегулярных запросов (например, ежедневных отчетов) [35].

Обработку данных осуществляют совместно *Apache Hadoop* и *HP Vertica*. Их ключевое различие — это тип данных, с которыми они работают лучше всего. *Hadoop* — программная платформа для выполнения распределенной обработки данных — подходит для задач, связанных с неструктурированными данными, такими как контент на естественном языке.

HP Vertica работает со структурированными данными, загруженными в таблицы БД. Применение этих двух платформ одновременно позволяет воспользоваться преимуществами каждой из них. Например, можно поручить *Hadoop MapReduce* извлечение и обработку ключевых слов из массы неструктурированных текстовых сообщений с сайта социальной сети, превращение материала в структурированные данные и затем загрузку их в БД *HP Vertica*. После загрузки можно выполнять множество различных аналитических запросов по данным значительно быстрее, чем в случае использования только *Hadoop*.

Обе эти платформы — *Hadoop* и *HP Vertica* — разделяют некоторые общие функции — используют кластеры хостов для хранения и работы с большими наборами данных. Коннектор, предоставляющий обмен данными между *Hadoop* и *HP Vertica*, запускается на каждом узле кластера *Hadoop* и поэтому его узлы и узлы *HP Vertica* взаимодействуют напрямую. Прямые соединения позволяют передавать данные параллельно, что значительно увеличивает скорость обработки. Коннектор написан на *Java* и совместим со всеми платформами, поддерживаемыми *Hadoop*.

Масштабирование кластеров Hadoop и HP Vertica. Узлы в кластере *Hadoop* подключаются непосредственно к узлам *HP Vertica* при получении или хранении данных. Если кластер *Hadoop* больше кластера *HP Vertica*, параллельная передача данных может негативно повлиять на производительность БД *HP Vertica*. Хорошее эмпирическое правило заключается в том, что кластер *Hadoop* не должен быть больше кластера *HP Vertica* [35, 36].

Существуют другие компоненты экосистемы *Apache Hadoop — Hive, HCatalog, WebHCat* и другие в составе *HP Vertica*. Так, *Hive* позволяет запрашивать данные, хранящиеся в распределенной файловой системе *Hadoop* (*HDFS*), используя набор классов *serializer* и *deserializer* (*SerDe*) для извлечения данных из файлов и разбиения их на столбцы и строки. Каждый *SerDe* обрабатывает файлы данных в определенном формате. Например, один *SerDe* извлекает данные из раз-

[6] *JSON* (*JavaScript Object Notation*) — текстовый формат обмена данными широко принят разработчиками веб-приложений. В сравнении с *XML* он легче в чтении и написании для людей и проще для анализа и генерации для ЭВМ.

[7] *CSV* (*comma-separated values* — значения, разделённые запятыми) — текстовый формат, предназначенный для представления табличных данных.

[8] *Understanding Flex Tables* — https://my.vertica. com/docs/7.0.x/HTML/Content/Authoring/FlexTables/Understanding FlexTables.htm

деленных запятыми файлов данных, а другой интерпретирует данные, хранящиеся в формате *JSON*. Консоль *HCatalog HP Vertica* предоставляет метаданные *Hive* доступными для других компонентов *Hadoop* (таких как *Pig*), позволяет получать доступ к хранилищу данных *Apache*, аналогично тому, как в родной таблице *HP Vertica*, и мн. др.

Таким образом, несмотря на признание огромной потенциальной ценности больших данных, часто может быть особенно сложно найти шаблоны или изучить новые формы полуструктурированных данных, генерируемых социальными сетями, веб-журналами, датчиками, текстовыми файлами с разделителями и пр. Эти новые, быстро растущие и критически важные типы данных обычно нуждаются в длительном процессе загрузки в традиционные аналитические платформы и хранилища данных, прежде чем они принесут ожидаемые результаты. Желание как можно скорее пролить свет на эти «темные» (*dark data*) данные, чтобы исследовать, проанализировать, понять их потенциал и при этом исключить необходимость интенсивного преобразования схем, которые в противном случае должны быть определены и применены до того, как данные будут загружены для исследования. Концепция *ELT* (извлечение, загрузка и преобразование) стала возможной и распространенной вследствие появления экосистемы продуктов *Hadoop*, *Hive*, *Pig* и активно используется в разработке *HP Vertica*. Возможности *one-step schema*™ позволяют создавать схемы загрузки в реляционные БД и использовать их как необходимые для высокопроизводительной аналитики и легко справляться с постоянно изменяющейся структурой данных.

Flex Zone взяла на себя неструктурированные, или нечеткие (*of schema-less data*), иначе, «темные» данные без посредника *NoSQL*. *Flex Zone* накладывает минимальное структурирование, в основном интерпретируя необработанные данные как ряд пар *ключ–значение* (основная структура данных большинства БД *NoSQL*). Оттуда необработанные данные могут запрашиваться с помощью *SQL*, либо напрямую, либо через инструменты *BI* и отчетов. Это позволяет получить полезное, хотя и неточное, прочтение данных.

Поскольку данные хранятся в колонках, а не в строках, отсутствующие значения столбцов не занимают места. *Flex Zone* эффективно использует архитектуру хранилища столбцов *Vertica MPP* и преобразует ее в структуру БД *NoSQL* семейства колоночных СУБД, такой как, например *HBase* [35, 36].

Распространенная в настоящее время *HPE Vertica* (*Hewlett Packard Enterprise*, *HPE's*) базируется на ядре *Vertica DBMS* (СУБД) и предлагает интеграцию с *Hadoop* для аналитики — *SQL* на *Hadoop*. В стеке *Hadoop* использован фреймворк *Apache Spark* с интегрированным в него модулем *Spark SQL* [37] на основе оптимизатора *Catalyst* [34, 38]. *Spark SQL* — стандартная библиотека, находящаяся поверх ядра *Spark-core*, — предназначена для структурированной обработки данных и реляционных запросов с использованием как *SQL*, так и специального *API DataFrame*. Они обеспечивают общий способ доступа к различным источникам данных, например, классическим реляционным СУБД, инфраструктуре хранилища (*schema on read*) данных *Hive* и других, а также интеграцию *SQL*-запросов со всеми элементами фреймворка *Spark*. *DataFrame* — распределенная коллекция данных, организованных в поименованных колонках. Это концептуально эквивалентно таблице в реляционной БД или кадру данных в *R*/*Python*. Для пользователей *DataFrame API* облегчает программирование *Spark SQL*.

DataFrames могут быть построены из широкого спектра источников: структурированных файлов данных, таблиц во внешних базах данных или существующих наборов данных *RDDs* (*resilient distributed datasets*). Источниками данных также могут служить хранилища сырых данных в исходном формате *Data Lake* [9] и тра-

[9] *Data Lake* «озеро данных» — хранилище сырых, необработанных данных разных форматов из различных источников, полученных в ходе исследования, до их востребования потребителем. Это обходится значи-

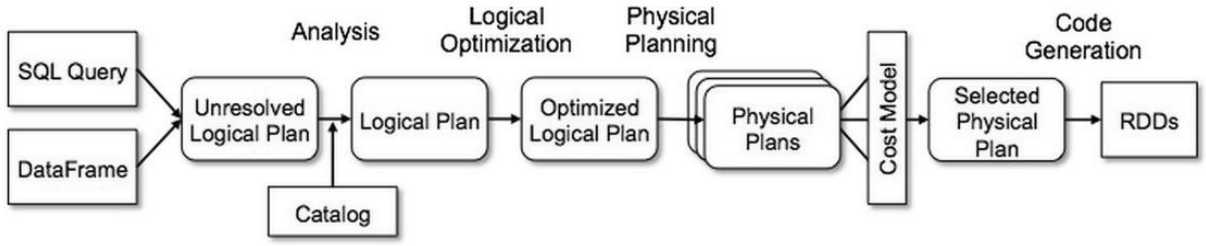

Рис. 1

диционные корпоративные хранилища данных — *Enterprise Data Warehouse* (*EDW*). Стандарт подключения к данным на сторонних носителях: *JDBC* или *ODBC*. По этим же интерфейсам обеспечивается соединение для инструментов бизнес-аналитики (*BI*). *HPE Vertica* поддерживает *BI* основных ведущих производителей и инструменты визуализации, а также встроенные аналитические функции. *DataFrame API* доступен в *Scala*, *Java*, *Python* и *R* [39].

Оптимизатор *Catalyst* [34, 38, 40] основан на конструкциях функционального программирования в *Scala* и использует расширенные возможности языка программирования (например, сопоставление шаблонов *Scala*) в новом способе оптимизации запросов. Его расширяемый дизайн имеет две цели: упростить добавление новых методов и функций оптимизации в *Spark SQL*, особенно для решения различных проблем, наблюдаемых с большими данными (например, полуструктурированными данными и расширенной аналитикой); обеспечить возможность расширения оптимизатора сторонними разработчиками, например, путем добавления специальных правил об источнике данных, которые могли бы вытеснить фильтрацию или агрегацию во внешние системы хранения или поддерживать новые типы данных [40].

Catalyst поддерживает оптимизацию на основе правил и затрат (рис. 1). Имеются библиотеки, специфичные для обработки реляционных запросов (например, выражения, планы логических запросов) и несколько наборов правил, обрабатывающих разные этапы выполнения запроса: анализ, логическую оптимизацию, физическое планирование и генерацию кода для компилирования части запросов к байт-коду *Java*. *Catalyst* предлагает несколько открытых точек расширения, внешние источники и пользовательские типы данных.

Другой оптимизатор — *Tungsten* [41] направлен на существенное повышение эффективности памяти и процессора для приложений *Spark*, чтобы приблизить производительность к пределам современного оборудования. Эти усилия предусматривают три инициативы:

- управление памятью: использование семантики приложений для явного управления памятью и устранения накладных расходов объектной модели *JVM* путем тонкой настройки сбора мусора в *Java*;
- *cache-aware* вычисления: алгоритмы и структуры данных построены так, чтобы использовать знания об иерархии памяти;
- генерация кода: динамическая генерация кода для использования современных компиляторов и процессоров вместо того, чтобы пошагово проходить медленный интерпретатор каждой строки.

По мнению аналитиков *Gartner* [23, 42], *HPE Vertica* ориентируется на основные тенденции рынка, поддерживая платформу для анализа больших данных в облаке, логические хранилища *LDW* (с *Vertica SQL* на *Hadoop*) и широкие возможности аналитики: проекты по машинному обучению, статистическому анализу и обработке графов; широкие возможности и функции *SQL*, анализ тональности текста, прогнозный и геопространственный анализ, сопоставление шаблонов последователь-

тельно дешевле традиционных хранилищ, в которые помещаются только структурированные данные.

ности событий, расширенные временные ряды и многое другое. Это привело к тому, что HP практически пересек границу и занял место внизу квадранта «Лидеры» *Magic Quadrant Gartner* 2014 г. (рис. 2) для систем управления хранилищами данных [42]. Управление данными для аналитики основывается на концепции *HAVEn* (Хейвене), сочетающей множество решений в аналитике под одним именем. *HPE Haven on demand* предлагает перспективный набор инструментов управления облачными данными и аналитические услуги, но отдельно от размещения *Vertica* [43, 44], что демонстрирует фрагментированную стратегию между этими двумя решениями.

Как отмечает *Gartner* [23, 42], *HPE Vertica* широко используется для различных случаев применения и типов данных, отличается от других представленных решений быстрым ответом на запрос.

Kdb. Общая характеристика. *Database* (*DB*) компании *Kx Systems* [45], ориентированная на индустрию финансовых услуг, основана в 1993 г. для решения одной из основных проблем высокопроизводительных вычислений, возникших в связи с экспоненциальным ростом объемов данных и неспособностью традиционной технологии реляционных БД обеспечить требуемые быстродействие и производительность в этой сфере. Классические реляционные СУБД, по мнению разработчика, не могут эффективно принимать и сохранять миллионы записей в секунду, не используют специальные подходы к обработке упорядоченных по времени данных.

Технология *Kx* изначально была воплощена в высокопроизводительной колоночно-ориентированной БД с использованием собственного языка программирования *K* [46]. Этот язык реализован на парадигмах матричного и функционального программирования, что обеспечивает компактную и быструю обработку массивов данных. *Kx* расширяет наследие векторных языков программирования и его функциональность [45, 46]. Ориентирован для работы с математическим анализом и финансовым прогнозированием, предназначен для

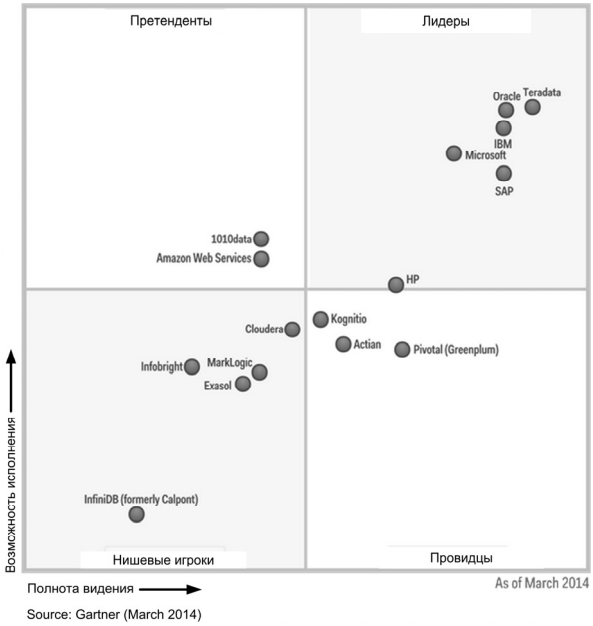

Рис. 2

работы с базами данных и создания финансовых приложений. Позднее *Kx* дополнительно вводит БД исторических данных и выпускает 64-битную версию[10] базы *kdb* под названием *kdb+*, включающую в себя язык *Q*, который объединил возможности *K* и *ksql* — *SQL*-подобного языка запросов. В настоящее время распространяется только язык *Q*. С другой стороны, собственный встроенный язык *Q* — проприетарный, коммерциализированный как и сама *kdb+*.

Обе базы относятся к типу *in-memory database, IMDS* (система баз данных в оперативной памяти) [47]. *Kdb+* подходит к обработке данных в памяти и на диске с единых позиций. Такая же архитектура используется для данных в реальном времени и ретроспективных данных, обеспечивая вычисления в оперативной памяти с высокой производительностью. Главное ее преимущество — обработка данных больших объемов. Допускается вертикальное и горизонтальное масштабирование: *Kx* способна работать на кластерах, выполненных на

[10] 32-разрядная версия доступна для некоммерческого применения.

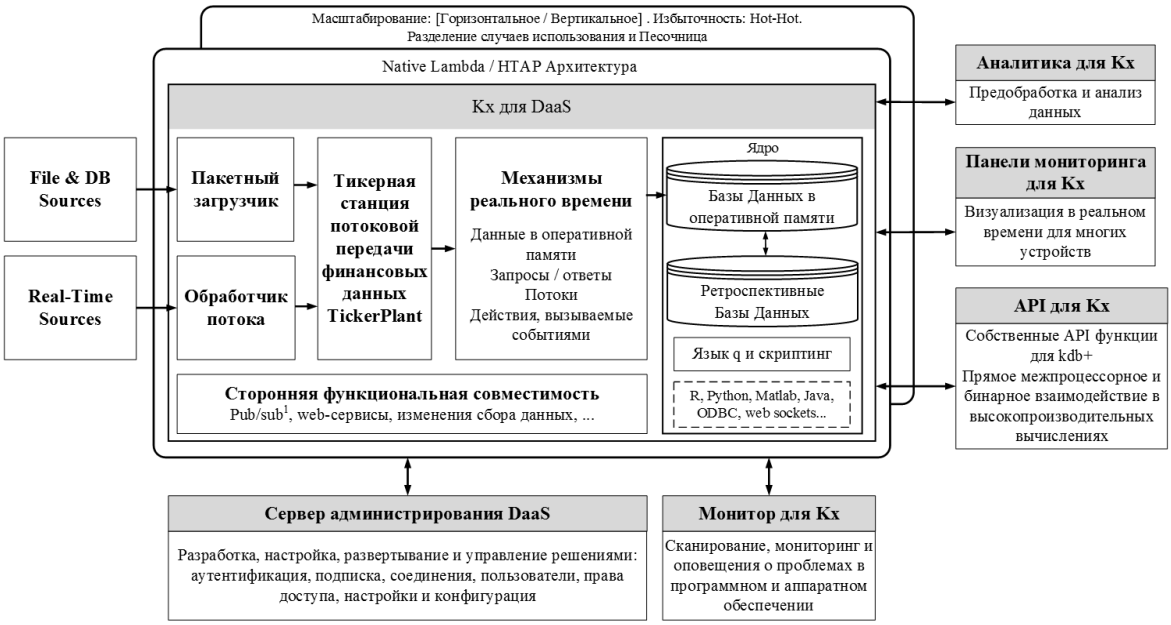

Рис. 3

стандартных серверных платформах на основе современных аппаратных многоядерных архитектур, *Grid*-сетях, на приложениях, исполняемых на облачных инфраструктурах. Дизайн обеспечивает плавную масштабируемость с ростом объемов данных до петабайт и выше без потери производительности.

Сочетание архитектурных и программных решений существенно упростило быстрый анализ временных рядов, характеризуемых высокой скоростью поступления. Встроенная поддержка операций временного ряда повысила скорость выполнения и гибкость запросов, агрегации, объединений и анализа деловой информации — структурированной или полуструктурированной. Работа ведется непосредственно с данными, устраняя необходимость отправки результатов запроса другим приложениям для анализа. Решение *Kx* обеспечило наилучшие допустимые задержки обработки даже с предлагаемыми *API*-интерфейсами для прямой поддержки сложной аналитики при высокоскоростных потоках входных данных.

Использование мультипарадигмального языка *Q* удешевило разработку *DB Kdb+*. Дистрибутив *kdb+* полностью, вместе с интерпретатором *Q* и примерами, имеет чрезвычайно малый размер и занимает всего несколько сотен килобайт, что упрощает установку и обслуживание. *DB Kdb+* доступна для *Solaris*, *Linux* и *Windows* [45].

Новые технологии обработки. В марте 2017 г. *Kx Systems* объявила о выпуске новой версии (*kdb+ ver.* 3.5.) своей базы данных и технологий аналитики. В это же время рассматривалось соглашение о крупномасштабной обработке геопространственных данных с использованием платформы аналитики *Kx* [45].

За годы развития и продвижения технологии клиентами *Kx Systems*, несмотря на дороговизну *kdb+*, стали крупнейшие биржи и финансовые учреждения, инвестиционные банки и компании мира, другие отрасли промышленности за пределами финансов, в том числе телекоммуникационные, фармацевтические, нефти и газа, софтверные компании и др. [45]. Все эти учреждения стремятся собирать, хранить, обрабатывать, распространять и, в конечном счете, монетизировать свои данные. Однако нет уверенности, что эти разобщенные данные могут быть использованы различными системами в дальнейшем. Причина в

том, что многие сохраняют даже внутри одного предприятия отдельные изолированные области данных. Это не только увеличивает общие затраты, вызванные поиском и сохранением множества копий данных, но также создает риск несогласованности. В то же время, данные должны быть постоянно доступны, легко потребляемы с гарантированно вносимыми исправлениями, что значительно ускорило бы разработку приложений.

Поэтому в настоящее время *Kx Systems* позиционирует свое решение *Kx for DaaS* как платформу предоставления данных. *DaaS* (*Data as a Service*) — динамическая доставка, или услуга предоставления данных по запросу. Решение *Kx* — это программный дизайн, объединяющий весь опыт и передовую технологию *Kx*, разработанную для финансовых рынков, в современную платформу для сбора данных и аналитики в реальном времени с предоставлением полного набора инструментов для управления данными с момента их приема до потребления несколькими сторонами согласованным, контролируемым образом [48]. Техническая архитектура *Kx for DaaS* приведена на рис. 3.

Следует обратить внимание на надпись «*Native Lambda / HTAP* Архитектура» во втором слое рисунка. *Gartner* [49, 50] объясняет *HTAP* (*hybrid transaction/analytical processing*) как *гибридную транзакционную / аналитическую обработку*, которые понимают как возможность одновременно выполнять эту противоречивую, с точки зрения ориентации на эти принципиально различные задачи, требующие отличных подходов к решению, формируемым структурам данных и видам нагрузок на вычислительные ресурсы [18, 20, 30], обработку записей в одной и той же базе данных. В настоящее время это возможно при соблюдении определенных условий.

Транзакционные системы (рис. 4), обычно операционные СУБД (оперативная обработка транзакций — *OLTP*), как правило, отделены от аналитических (*OLAP*) и основаны на разных архитектурах, обусловленных задачами и функциями обработки. С одной стороны, такой подход позволил объединить данные нескольких источников в хранилище, с другой — данные, нормализованные для производительности транзакционной обработки, должны быть извлечены из операционной БД, преобразованы и загружены в аналитические БД — хранилища, витрины данных для поддержки аналитики. Но последнее приводит к задержке аналитической обработки (аналитическая задержка), что может занять часы, дни или даже недели с момента обработки данных приложением транзакций до времени, когда они могут использоваться для аналитики. Аналитическая задержка, в свою очередь, в операциях развертывания агрегированных данных зачастую приводит к рассинхронизации извлекаемых детализированных данных относительно исходных данных в операционной БД, используемых для транзакционной обработки — *TP* (*transaction processing*).

Таким образом, разделение транзакционной обработки и аналитики, выполнение их на разных архитектурах усложняют информационную архитектуру и соответствующую инфраструктуру, а также привносит задержку в

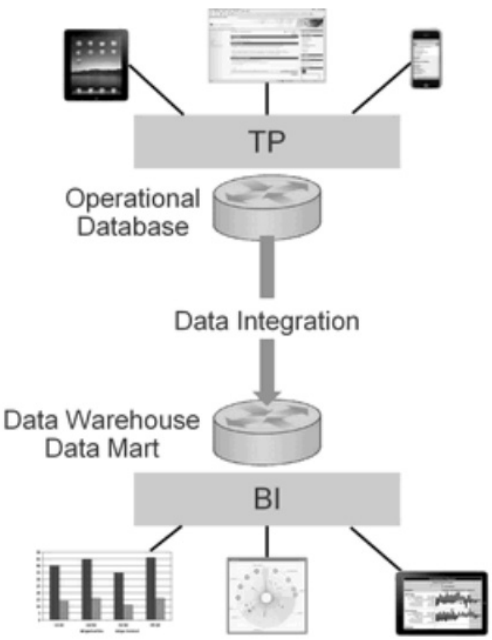

Рис. 4

анализ данных. Благодаря технологическим достижениям, таким как вычисления в памяти (*in-memory computing*, *IMC*), стала возможной архитектура гибридной транзакционно/аналитической обработки — *HTAP*, позволяющая приложениям анализировать «живые» данные по мере их создания и *обновления функциями обработки транзакций*. Это означает, что для аналитики в режиме реального времени запрашиваются данные, собранные непосредственно из *необработанных* транзакционных данных. При этом становится возможным расширенная аналитика для принятия решений в режиме реального времени с использованием гибридной потоковой передачи и вычислений в оперативной памяти — *IMC* — для решения задач аналитики и обработки транзакций [49].

Расширенная аналитика в режиме реального времени, такая как прогнозирование и моделирование, становится неотъемлемой частью самого процесса, а не позиционируется как отдельное действие, выполненное после. Это позволит в рамках действующих процессов выполнять гораздо более совершенный и сложный анализ их бизнес-данных в режиме реального времени, чем с использованием традиционной архитектуры. Более сложная диагностическая и прогнозная аналитика в режиме реального времени позволит принять решения о тенденциях и ситуациях в бизнесе, требующих немедленного реагирования.

При введении в систему исторических данных (*historical database*, *HDB*) они совмещаются с потоковыми данными реального времени (*real-time database*, *RDB*[11]), при необходимости масштабируются [49] и подвергаются обработке в оперативной памяти, что обеспечивает динамическое объединение данных разных временных измерений и их визуализацию в табличной и графической формах в режиме реального времени. При этом различные независимые аналитические и *TP*-приложения совместно используют одни и те же наборы данных в памяти, а их методы обработки переплетены в одной и той же поддерживающей вычисления памяти (*IMC-Enabled*). *RDB* в оперативной памяти отвечает за бизнес-потребности реального времени, в то время как *HDB* является неизменяемой записью всех предыдущих транзакций. Выполнив запрос к *RDB* и *HDB*, всегда можно получить последовательное представление о мире и состоянии бизнеса. *HTAP* предоставит аналитикам возможность «жить в своих данных», одновременно взаимодействуя как с историческими, так и потоковыми данными в реальном времени.

Эта широко используемая архитектура *RDB/HDB* была принята сообществом *Big Data*. Изменения в модели описаны как *Архитектура Lambda*, *гибридная транзакционная/аналитическая обработка* (*HTAP*) *Gartner*. Эта гибридная архитектура быстро развивается как инновационная архитектура обработки и приложений, поддерживаемых вычислениями в оперативной памяти (*IMC*).

Устранив проблемы с аналитической задержкой и синхронизацией данных, *HTAP* позволит упростить инфраструктуру управления информацией, по крайней мере, за счет дублирования и путем согласованности данных. В традиционной архитектуре контроль и управление несколькими копиями одних и тех же данных должны поддерживаться согласованно, в противном случае это может привести к неточностям, временным различиям и несогласованности данных.

Вместе с тем, имеются проблемы, связанные с использованием технологии *IMC* — ключевым фактором реализации инициатив в области *HTAP*: код традиционных приложений, как правило, должен быть переработан, чтобы максимально использовать технологию *IMC* и обеспечить интеграцию с расширенными инструментами аналитики; переход на версию традиционного пакетного приложения, поддерживающего работу с *IMC*, необязательно приводит к архитектуре *HTAP*, если

[11] Все чаще *RDBs* размещаются в огромной энергонезависимой памяти, а *HDBs* хранятся на быстрых *SSD* [49].

не пройден процесс оптимизации кода продукта для *IMC* и не добавлена аналитика в режиме реального времени и др.

Индустрия финансовых услуг успешно справлялась с аналогичными проблемами в течение более двух десятилетий. Банки и торговые фирмы использовали простую масштабируемую архитектуру данных, состоящую из базы данных реального времени (*RDB*) и базы данных истории (*HDB*). Современные предприятия нуждаются в быстром интерактивном доступе к десяткам или сотням терабайт данных, извлекаемых из огромных озер и устаревших систем данных. Они также требуют совмещения нескольких источников данных как исторических, так и получаемых в реальном времени, для динамического объединения данных разных измерений и их визуализации в табличной и графической форме. Аналитики, работающие с данными, должны иметь возможность «жить в своих данных», одновременно взаимодействуя как с историческими, так и с потоковыми данными в реальном времени.

Внутрисхемная *RDB* адресует бизнес-потребности в реальном времени, в то время как *HDB* является неизменной записью всех предыдущих транзакций. Посредством запросов к *RDB* и *HDB* всегда можно получить целостное представление о мире и состоянии бизнеса. Подход основан на гибриде потоковой передачи, *RDB* внутрисхемных вычислений (*IMC*) и неизменных записей *HDB*. Эффективная работа с современными решениями памяти позволяет запрашивать все свои данные в режиме, близком к реальному времени, обеспечивающем целостную картину, собранную непосредственно из исходных данных требуемых транзакций [49, 50].

Gartner [49] полагает, что архитектуры хранилищ данных останутся необходимыми для поддержки расширенного анализа, который содержит большой объем исторических данных или Больших Данных, поступающих из множества внутренних и внешних, структурированных и неструктурированных источников. Отдельные системы *HTAP* будут вносить вклад в логические или физические хранилища данных, но не будут полностью их заменять.

Заключение

Сопоставив изложенное с архитектурным решением *Kx Systems* (*kdb+*), видим подобие в реализации ключевых технологий *IMC* и слиянии потоков *RDB* и *HDB*. Принимая во внимание, что стоящее первым в выражении «*Native Lambda / HTAP* Архитектура» слово подчеркивает исконное свойство, изначально заложенное в архитектуру разработки *Kx Systems*, можно быть уверенным, что концепция *HTAP Gartner* основана на реалиях и *kdb+* здесь не на последнем месте. Так, она допускает создание простых форм *HTAP*-приложений с использованием традиционных СУБД и считает, что большинство реализаций *HTAP* должны и будут поддерживать *IMC*. Технологии *IMC*, такие как СУБД в оперативной памяти (*IMDBMSs*) и высокомасштабируемые, отказоустойчивые хранилища данных (ХД) в оперативной памяти *Grid*-среды — *in-memory data grids*[12] (*IMDGs*), поддерживают единое ХД с доступом к его памяти с низкой латентностью, которое может обрабатывать большие объемы транзакций. Эти технологии также могут поддерживать текущую аналитику с нулевой задержкой в отношении тех же самых данных, включая расширенную аналитику, такую как прогнозирование и моделирование, а также более традиционные стили описательного анализа [49].

Дальнейшее исследование материала будет представлено в последующих публикациях.

[12] *IMDGs* — тип распределенного программного хранилища, отличного от реляционной БД в памяти, БД *NoSQL* или реляционной СУБД. Модель данных не является реляционной или объектно-ориентированной. Структура данных хранилища – ключ / значение. Данные распределены по многим серверам кластера и хранятся в их оперативной памяти. Хранилище отличается высокой производительностью [51].

СПИСОК ЛИТЕРАТУРЫ

30. *Gritsenko V.I.*, *Oursatyev A.A.* Big Data and the Tools for Analytics, Upr. sist. maљ., 2017, N 4, P. 3–14. (In Russian).
31. *Hinchcliffe Dion*. The enterprise opportunity of Big Data: Closing the "clue gap", http://www.zdnet.com/article/the-enterprise-opportunity-of-big-data-closing-the-clue-gap/
32. *HP Vertica*, http://www.vertica.com/
33. *IT architect* of the data warehouse architect. The choice of Vertica VS, Jan. 28, 2013, http://ascrus.blogspot.com /2013/01/vertica-vs.html (In Russian).
34. *Borchuk L.* Value Optimizers for DBMS: yesterday and today. Open Systems, 2016, N 1, P. 36–39. (In Russian).
35. *HP Vertica* Analytics Platform Version 7.0.x Documentation. Flex Zone, https://my.vertica.com/docs/7.0.x/HTML/index.htm#Authoring/FlexTables/FlexTableHandbook.htm%3FTocPath%3DFlex%2520Tables%2520Guide%7C_0
36. *Brust Andrew*. Vertica 7 to NoSQL DBs: Drop dead. ZDNet — for Big on Data, 21 Nov. 2013, Topic: Big Data Analytic, http://www.zdnet.com/article/vertica-7-to-nosql-dbs-drop-dead/
37. *Spark* SQL: Relational Data Processing in Spark / M. Armbrust, R. Xin, C. Lian et al., Proc. of the 2015 ACM SIGMOD Int. Conf. on Management of Data, 31 May — 4 June 2015, Melbourne, Victoria, Australia, 2015, http://people.csail.mit.edu/matei/papers/2015/ sigmod_spark_sql.pdf
38. *Vertica Blog*. Looking Under the Hood at Vertica Queries, 02 Mar. 2016, https://my.vertica.com/blog/ looking-under-the-hood-at-vertica-queriesba-p235038/
39. *Spark SQL* and DataFrames. Spark 1.5.2 Documentation, http://spark.apache.org/docs/latest/sql-programming-guide.html
40. *Deep* Dive into Spark SQL's Catalyst Optimizer. M. Armbrust, Y. Huai, C. Liang et al., 15 Apr. 2015, https://databricks.com/blog/2015/04/13/deep-dive-into- spark-sqls-catalyst-optimizer.html
41. *Xin R.*, *Rosen J.* Project Tungsten: Bringing Apache Spark Closer to Bare Metal, 28 Apr. 2015, https://databricks.com/blog/2015/04/28/project-tungsten-bringing-spark-closer-to-bare-metal.html
42. *Mark A. Beyer*, *Edjlali R.* Magic Quadrant for Data Warehouse Database Management Systems, 7 Mar. 2014, https://www.slideshare.net/paramitap/gartner-magic-quadrant-for-data-warehouse-database-management-systems
43. *HP Haven* OnDemand, http://www8.hp.com/ua/ru/software-solutions/big-data-cloud-haven-ondemand/
44. *Платформа* для больших объемов данных, http:// www8.hp.com/ua/ru/software-solutions/big-data-platform-haven/
45. *Kx*, https://kx.com
46. *Encyclopedia* of programming languages. K (programming language), http://progopedia.ru/language/k/ (In Russian).
47. *Graves Steve*. In-Memory Database Systems, 1 Sep. 2002, http://www.linuxjournal.com/article/6133
48. *Gartner*. Delivering Scalable and Robust Data Infrastructures with DaaS in Financial Markets. Kx for DaaS, Feb. 2017, http://www.gartner.com/imagesrv/media-products/pdf/Kx/KX-1-3RU8DEE.pdf
49. *Gartner*. Real-time Insights and Decision Making using Hybrid Streaming, In-Memory Computing Analytics and Transaction Processing, https://www.gartner.com/imagesrv/media-products/pdf/Kx/KX-1-3CZ44RH.pdf
50. *Pezzini Massimo*. Predicts 2016: In-Memory Computing-Enabled Hybrid Transaction/Analytical Processing Supports Dramatic Digital Business Innovation, Jan. 2016, https://www.linkedin.com/pulse/predicts-2016-in-memory-computing-enabled-hybrid-supports-pezzini
51. *Colmer P.* In Memory Data Grid Technologies Wednesday, 21 Dec. 2011, http://highscalability.com/blog/ 2011/12/21/in-memory-data-grid-technologies.html

Поступила 16.01.2018

A.A. Oursatyev, PhD in Techn. Sciences, Leading Research Associate, 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 WAREHOUSE: VERTICA, KDB

Introduction. The article is a continuation of the research on the Great Data and the toolkit that transforms into a new generation of technologies and architectures of databases platforms and Warehouse for the intelligent output. The progressive industrial developments of the world-famous IT companies, including some specialized hardware-software storage and processing systems described in *Hinchcliffe Dion*, are presented. The opportunity of Big Data: the closing of the "clue gap" are presented, the changes in the infrastructure, tools and platforms for obtaining the necessary information and new knowledge of the Big Data are analysed. The material is presented in such a way that the focus is on the transformation of a known database environment, databases or Data Warehouse for intelligent output, and initial information about a specific product is given in the general product characteristics. The second part of the review is represented by the database *Vertica and Kdb*.

Purpose. It is necessary to consider the infrastructure solutions for the new analytics-oriented developments and evaluate the effectiveness of their application in the studies of the Great Data for new knowledge, the discovery of implicit connections and in-depth understanding.

Methods. Information-analytical methods and technologies of data processing, the methods of their estimation and forecasting, taking into account the development of the most important branches of informatics and information technologies.

Results. The column-oriented *DBMS Vertica*, based on the *MPP* architecture, is designed to work in a horizontally scalable environment.

From version 7.0, the *HP Vertica Analytics Platform* integrated into its platform ecosystem products consisting of *Apache Hadoop* and a number of the additional modules, which allowed to create a special area of storage and processing - *Flex Zone*, based on flex or flexible tables. *Flex Zone* has assumed the unstructured or fuzzy (of schema-less data) data without the *NoSQL* intermediary. *Flex Zone* imposes minimal structuring, basically interpreting raw data as a series of key-value pairs. From it the raw data can be requested using *SQL*, either directly, or through *BI* tools. It provides useful, though inaccurate, reading of data. *HPE Vertica* is currently based on the *Vertica DBMS* core and offers the integration with *Hadoop* for analytics - *SQL* for *Hadoop*. The *Hadoop* stack uses the *Apache Spark* framework with the *Spark SQL* based on its *Catalyst* and *Tungsten* optimizers. The first supports cost-based optimization, the second-*Tungsten* aims to increase memory efficiency and processor performance for *Spark* applications to bring performance up to the limits of state-of-the-art equipment.

Kdb High-performance column-oriented *DB* company *Kx Systems* uses its own programming language *K*. This language is implemented on the paradigms of matrix and functional programming, which provides a compact and fast processing of data arrays. It is Oriented to the work with mathematical analysis and financial forecasting, and it is designed for databases and financial applications. Later, *Kx* additionally inserts a database of historical data and releases a 64-bit version of the *kdb* database called *kdb* +, which includes the language *Q*, which combines the capabilities of *K* and *ksql* - *SQL*-like query language. Currently only *Q* is propagated and commercialized as *kdb* +.

The database belongs to the type of in-memory database. *Kdb* + is suitable for data processing in memory and on a single position disk. The same architecture is used for real-time data and retrospective data.

At present, *Kx Systems* proposes its *Kx* for *DaaS* solution as a dynamic delivery or data provision service upon request. The *Kx* solution is a modern real-time data and analytics platform providing a set of tools for managing data from the moment they are received to the consumption by several parties, coordinated and controlled.

The *Kx* solutions are discussed in comparison with the *HTTP Gartner architecture*, which allows the applications to analyze data directly from their receipt and update the functions of transaction processing. In this case, it is possible both to transact the processing, and to expand analyst, to make decisions in real time using hybrid streaming and computing in RAM. Analysis - forecasting and modeling, becomes an integral part of the process, rather than being positioned as a separate action.

Conclusion. According to analysts, *Gartner HPE Vertica* focuses on the main market trends, supporting the analysis of large data in the cloud, logical Warehouse *LDW* (from *Vertica SQL* to *Hadoop*). Data management offers a promising set of tools, but they are separated from the placement of *Vertica*, demonstrating a fragmented strategy between these two solutions. As Gartner points out, *HPE Vertica* is widely used for different application cases and types of data and is different from other submitted solutions by fast response to a request.

Gartner believes that individual *HTAP* systems will contribute to logical or physical storage, but will not completely replace them. At the same time, the data warehousing architecture will remain necessary to support the extended analysis that contains a large amount of historical data or large data coming from internal and external, structured and unstructured sources.

Keywords: *MPP — architecture, HTAP — hybrid transactional/analytical processing, LDW — logical data warehouse, cloud storage, database platform as DBPaaS service, SaaS model analyst, data management environment, IMC technology.*

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

ВЕЛИКІ ДАНІ. АНАЛІТИЧНІ БАЗИ ДАНИХ І СХОВИЩА: *VERTICA, KDB*

Вступ. Стаття є продовженням досліджень Великих Даних і інструментарію, що трансформується в нове покоління технологій і архітектури платформ баз даних і сховищ для інтелектуального висновку. Розглянуто прогресивні промислові розробки відомих у світі IT-компаній, зокрема деякі спеціалізовані апаратно-програмні системи зберігання і обробки, наведені в роботі *Hinchcliffe Dion. The enterprise opportunity of Big Data*: *Closing the* "*clue gap*", та проаналізовано зміни інфраструктури, інструментального середовища і платформи для отримання необхідної інформації та нових знань з Великих Даних. Матеріал подано так, що основну увагу приділено питанням

трансформації відомого середовища БД, СУБД або сховищ даних (*Data Warehouse*) для інтелектуального висновку, а початкові відомості про конкретний продукт подано у загальній характеристиці виробу. У другій частині огляду представлено БД *Vertica* і *Kdb*.

Мета. Досить доцільно розглянути інфраструктурні рішення нових розробок, орієнтованих на аналітику, і оцінити ефективність їх застосування в дослідженнях Великих Даних для отримання нових знань, виявлення неявних зв'язків і поглибленого розуміння, проникнення в сутність явищ і процесів.

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

Результати. Колоночно-орієнтовану СУБД *Vertica*, засновану на архітектурі *MPP*, призначено для роботи в горизонтально масштабованому середовищі і оптимізовано для запису та зберігання даних.

З версії 7.0 *HP Vertica Analytics Platform* інтегрувала в свою платформу екосистему продуктів, що складається з *Apache Hadoop* та додаткових модулів, що дозволило їй створити спеціальну область зберігання і обробки даних — *Flex Zone*, засновану на технологіях гнучких (*flex or flexible*) таблиць. *Flex Zone* взяла на себе неструктуровані або нечіткі (*of schemaless data)* дані без посередника *NoSQL* і накладає мінімальне структурування, в основному інтерпретуючи необроблені дані як пари «ключ-значення». З неї необроблені дані можна запитувати за допомогою *SQL* безпосередньо або через інструменти *BI*. Це дає корисне, хоча і неточне, прочитання даних. Поширена в даний час *HPE Vertica* базується на ядрі *Vertica DBMS* і пропонує інтеграцію з *Hadoop* для аналітики — *SQL* на *Hadoop*. У стеку *Hadoop* використано фреймворк *Apache Spark* з інтегрованим модулем *Spark SQL* на основі оптимізаторів *Catalyst* і *Tungsten*. Перший підтримує оптимізацію на основі правил і видатків, другий — *Tungsten* спрямований на підвищення ефективності пам'яті і процесора для додатків *Spark*, щоб наблизити продуктивність до меж сучасного обладнання.

Kdb. Високопродуктивна колоночно-орієнтована БД компанії *Kx Systems* з використанням власної мови програмування *K*. Ця мова реалізована на парадигмах матричного і функціонального програмування, що забезпечує компактну і швидку обробку масивів даних. Орієнтована для роботи з математичним аналізом і фінансовим прогнозуванням, призначена для баз даних і створення фінансових застосунків. Пізніше, *Kx* додатково вводить БД історичних даних і випускає 64-бітну версію бази *kdb* під назвою *kdb* +, що включає в себе мову *Q*, яка об'єднала можливості *K* і *ksql* — *SQL*-подібної мови запитів. Зараз поширюється тільки мова *Q* — пропріетарна, комерціалізована як і сама *kdb* +.

БД належить до типу *in-memory database. Kdb* + підходить до обробки даних в пам'яті і на диску з єдиних позицій. Така ж архітектура використовується для даних в реальному часі і ретроспективних даних.

В даний час *Kx Systems* позиціонує своє рішення як динамічна доставка або послуга надання даних за запитом. Рішення *Kx* — це сучасна платформа для збору даних і аналітики в реальному часі з наданням набору інструментів для управління даними з моменту їх прийому до споживання декількома сторонами узгоджено і контрольоване.

Обговорюються рішення *Kx* в порівнянні з *HTAP*-архітектурою *Gartner*, яка дозволяє застосуванням аналізувати дані безпосередньо за їх надходженням і оновленням функціями обробки транзакцій. При цьому стає можливим як обробка транзакцій, так і розширена аналітика, для прийняття рішень в режимі реального часу з використанням гібридної потокової передачі і обчислень в оперативній пам'яті. Аналітика — прогнозування і моделювання, стає невід'ємною частиною процесу, а не позиціонується як окрема дія.

Висновок. На думку аналітиків, *Gartner HPE Vertica* орієнтується на основні тенденції ринку, підтримуючи аналіз великих даних в хмарі, логічні сховища *LDW* (з *Vertica SQL* на *Hadoop*). Управління даними пропонує перспективний набір інструментів, але окремо від розміщення *Vertica*, що демонструє фрагментовану стратегію між цими двома рішеннями. Як зазначає *Gartner*, *HPE Vertica* широко використовується для різних випадків застосування і типів даних та відрізняється від інших поданих рішень швидкою відповіддю на запит.

Gartner вважає, що окремі системи *HTAP* зроблять внесок в логічні або фізичні сховища даних, але не будуть повністю їх замінювати. Разом з тим, архітектури сховищ даних залишаться необхідними для підтримки розширеного аналізу, який містить великий обсяг історичних або Великих Даних.

Ключові слова: *MPP — архітектура, HTAP — гібридна транзакційна/аналітична обробка, LDW — логічні сховища даних, хмарне зберігання, платформа баз даних як послуга DBPaaS, аналітика за моделлю SaaS, середовище управління даними, технологія IMC.*