ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 20547-3— 2024

Информационные технологии

ЭТАЛОННАЯ АРХИТЕКТУРА БОЛЬШИХ ДАННЫХ

Часть 3

Эталонная архитектура

(ISO/IEC 20547-3:2020, IDT)

Издание официальное

Москва
Российский институт стандартизации
2024

Предисловие

1 ПОДГОТОВЛЕН Научно-образовательным центром компетенций в области цифровой экономики Федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный университет имени М.В. Ломоносова» (МГУ имени М.В. Ломоносова) и Обществом с ограниченной ответственностью «Институт развития информационного общества» (ИРИО) на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 164 «Искусственный интеллект»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 28 октября 2024 г. № 1541-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 20547-3:2020 «Информационные технологии. Эталонная архитектура больших данных. Часть 3. Эталонная архитектура» (ISO/IEC 20547-3:2020 «Information technology — Big data reference architecture — Part 3: Reference architecture», IDT).

ИСО/МЭК 20547-3:2020 разработан подкомитетом SC 42 «Искусственный интеллект» Объединенного технического комитета ISO/IEC JTC 1 «Информационные технологии».

Дополнительная сноска в тексте стандарта, выделенная курсивом, приведена для пояснения текста оригинала.

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.rst.gov.ru)

Содержание

Введение

Серия стандартов ИСО/МЭК 20547 предназначена для предоставления пользователям стандартизованного подхода к разработке и внедрению архитектур больших данных и для предоставления необходимых справочных материалов. ИСО/МЭК 20547-1 содержит общие сведения о структуре эталонной архитектуры, представленной в стандарте, а также описывает процесс ее применения в ходе выполнения разработки архитектуры. ИСО/МЭК 20547-2 включает набор вариантов использования больших данных и описывает их в виде совокупности технических условий, которые могут быть учтены архитекторами больших данных и разработчиками систем, а также могут быть использованы архитектором больших данных для описания конкретной системы. В ИСО/МЭК 20547-4 рассмотрены аспекты защищенности и конфиденциальности персональных данных, которые являются уникальными для больших данных. ИСО/МЭК 20547-5 содержит перечень стандартов и описывает их взаимосвязи с эталонной архитектурой, которую архитекторы и разработчики могут рассматривать как составную часть процесса проектирования и реализации своей системы.

Каждая из этих частей построена на общих подходах: словаре и концепциях, представленных в ИСО/МЭК 20546.

В общих чертах эталонная архитектура представляет собой авторитетный источник информации о конкретной предметной области, который направляет и ограничивает реализацию нескольких архитектур и решений (см. 3.2).

Эталонные архитектуры обычно служат рекомендованной основой для архитектурных решений, а также могут быть использованы в целях сравнения и согласования.

Ключевая цель эталонной архитектуры — способствовать общему пониманию существующих архитектур и будущих направлений их развития в рамках множества продуктов, организаций и дисциплин.

Эталонная архитектура, описанная в настоящем стандарте, представляет собой структуру архитектуры, предназначенную для описания компонентов больших данных, процессов и систем больших данных, чтобы установить общий (универсальный) язык для общения заинтересованных сторон, и носит наименование эталонной архитектуры больших данных. Системная архитектура конкретной системы больших данных в настоящем стандарте не представлена: он является инструментом для описания и обсуждения, а также разработки системных архитектур с использованием структуры эталонной архитектуры и обеспечивает общие высокоуровневые архитектурные представления, которые служат эффективным средством обсуждения требований, структур и операций, присущих большим данным. Представленная в настоящем стандарте модель не привязана к продуктам, сервисам или эталонным реализациям конкретных поставщиков, а также не содержит нормативных решений, препятствующих инновациям.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

ЭТАЛОННАЯ АРХИТЕКТУРА БОЛЬШИХ ДАННЫХ

Часть 3

Эталонная архитектура

Information technology.
Big Data Reference architecture.
Part 3. Reference architecture

Дата введения — 2025—01—01

1 Область применения

Настоящий стандарт определяет эталонную архитектуру больших данных, которая включает концепции и архитектурные представления.

Эталонная архитектура больших данных, представленная в настоящем стандарте, определяет две архитектурные точки зрения:

- пользовательское представление, определяющее роли/подроли, их отношения и типы действий в экосистеме больших данных;
- функциональное представление, определяющее архитектурные уровни и классы функциональных компонентов на этих уровнях, которые реализуют виды деятельностей ролей/подролей в пользовательском представлении.

Эталонная архитектура больших данных предназначена:

- для обеспечения общего языка для различных заинтересованных сторон;
- для поощрения приверженности соблюдения общих стандартов, спецификаций и шаблонов;
- для обеспечения согласованности реализации технологии для решения однотипных наборов задач;
- для облегчения понимания операционных особенностей больших данных;
- для иллюстрации и понимания различных компонентов, процессов и систем в контексте общей концептуальной модели больших данных;
- для подготовки к представлению технической справки для государственных организаций, агентств и других потребителей, обеспечивающей возможности понимания, обсуждения, классификации и сравнения решений для больших данных;
- для взвешенного анализа создаваемых стандартов с точки зрения интероперабельности, переносимости, возможности повторного использования и расширяемости.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты [для датированных ссылок применяют только указанное издание ссылочного стандарта, для недатированных — последнее издание (включая все изменения)]:

ISO 8000-2, Data quality — Part 2: Vocabulary (Качество данных. Часть 2. Словарь)

Издание официальное

ISO/TS 8000-60, Data quality — Part 60: Data quality management: Overview (Качество данных. — Часть 60. Управление качеством данных. Обзор)

ISO 8000-61 Data quality — Part 61: Data quality management: Process reference model (Качество данных. Часть 61. Менеджмент качества данных. Эталонная модель процесса)

ISO/IEC 38500 Information technology — Governance of IT for the organization (Информационные технологии. Управление ИТ в организации)

ISO/IEC 38505-1, Information technology — Governance of IT — Governance of data — Part 1: Application of ISO/IEC 38500 to the governance of data (Информационные технологии. Управление ИТ. Управление данными. Часть 1. Применение ИСО/МЭК 38500 к управлению данными)

ISO/IEC TR 38505-2, Information technology — Governance of IT — Governance of data — Part 2: Implications of ISO/IEC 38505-1 for data management (Управление активами. Системы менеджмента. Руководство по применению ИСО 55001)

ISO 55000, Asset management — Overview, principles, and terminology (Управление активами. Обзор, принципы и терминология)

ISO 55001, Asset management — Management systems — Requirements (Управление активами. Системы менеджмента. Требования)

ISO 55002, Asset management — Management systems — Guidelines for the application of ISO 55001 (Управление активами. Системы менеджмента. Руководство по применению ИСО 55001)

ISO/IEC/IEEE 42010, Systems, and software engineering — Architecture description (Разработка систем и программного обеспечения. Описание архитектуры)

ISO/IEC 20546, Information technology — Big data — Overview and vocabulary (Информационные технологии. Большие данные. Обзор и словарь)

ISO/IEC 17789*, Information technology — Cloud computing — Reference architecture (Информационные технологии. Облачные вычисления. Эталонная архитектура)

3 Термины и определения

ИСО и МЭК поддерживают терминологические базы данных для использования в стандартизации, расположенные по следующим адресам:

- платформа ИСО для онлайн-просмотра материалов по стандартам (Online Browsing Platform, OBP), доступная по адресу: https://www.iso.org/obp/ui;
- база данных МЭК Электропедия (IEC Electropedia), доступная по адресу: http://www.electropedia.org/

В настоящем стандарте применены термины по ИСО 8000-2, ISO 8000-2, ISO/TC 8000-60, ИСО 8000-61, ИСО/МЭК 38500, ИСО/МЭК 38505-1, ИСО/МЭК ТР 38505-2, ИСО 55000, ИСО 55001, ИСО 55002, ISO/IEC/IEEE 42010, ИСО/МЭК 20546, ИСО/МЭК 17789, а также следующие термины с соответствующими определениями:

3.1 **данные** (data): Представление информации (3.3) в формальном виде, пригодном для передачи, интерпретации или обработки.

[ИСО/МЭК 2382:2015, 2121272]

3.2 **эталонная архитектура** (reference architecture): Авторитетный источник информации о конкретной предметной области, который направляет и ограничивает реализацию множества архитектур и решений.

Примечание 1 — В настоящем стандарте используется определение эталонной архитектуры из документа «Описание эталонной архитектуры» Министерства обороны США (DoD) [7].

Примечание 2 — Эталонные архитектуры обычно служат основой для выбора варианта архитектуры, а также могут быть использованы для сравнения и согласования конкретных вариантов архитектур и принятых решений.

3.3 **информация** (information): Данные (3.1), которые обрабатывают, организуют и коррелируют для получения выходного значения.

Примечание — Информация касается фактов, концепций, объектов, событий, идей, процессов и т. д.

* Заменен на ISO/IEC 22123-3:2023. Однако для однозначного соблюдения требования настоящего стандарта рекомендуется использовать только указанное в этой ссылке издание.

3.4 **деятельность** (activity): Заданная последовательность или совокупность задач.

[ИСО/МЭК 17789:2014, 3.2.1]

3.5 **знание** (knowledge): Сохраняемая, обрабатываемая и интерпретируемая информация (3.3).

[ISO 5127:2017, 3.1.1.17]

3.6 **функциональный компонент** (functional component): Функциональный структурный блок, необходимый для участия в виде деятельности (3.4), поддерживаемой в ходе имплементации.

[ИСО/МЭК 17789:2014, 3.2.3]

3.7 **стратегическое управление данными** (data governance): Свойство или способность, которые необходимо координировать и реализовывать с помощью набора действий (3.4), направленных на разработку, внедрение и мониторинг стратегического плана управления информационными активами.

Примечание 1 — Стратегическое управление данными описано в ИСО/МЭК 38505-1.

Примечание 2 — Под активом данных понимается набор элементов данных или объектов данных, которые имеют реальную или потенциальную выгоду для организации. Актив данных — это подмножество активов, определенных в ИСО 55000. Выгода — это преимущество организации практических знаний, полученных исходя из возможностей аналитической системы, которое относят к большим данным из-за понимания того, что данные имеют потенциальную пользу, которая ранее обычно не рассматривалась.

Примечание 3 — Стратегический план управления активами данных — это документ, который определяет, как управлять данными (3.15), и должен быть согласован со стратегией организации. Этот термин имеет такое же значение, как и план стратегического управления активами, определенный в ИСО 55000 с точки зрения данных.

3.8 **качество данных** (data quality): Свойство, определяющее степень, с которой набор характеристик, присущих данным, отвечает требованиям организации.

[ИСО 25024:2015, 4.11]

3.9 **управление качеством данных** (data quality management): Скоординированные действия по руководству и контролю организации в части обеспечения качества данных.

[ИСО 8000-2:2022, 3.8.2]

3.10 **сторона** (party): Физическое или юридическое лицо, зарегистрированное или незарегистрированное, или их группа.

[ИСО/МЭК 17789:2014, статья 7.2.3]

3.11 **политика** (policy): Намерения и курс организации, официально сформулированные ее высшим руководством.

[ИСО 55000:2014, 3.1.18, изменено — термин изменен на форму единственного числа, а окончание удалено из определения]

3.12 **роль** (role): Набор действий (3.4), которые служат общей цели.

[ИСО/МЭК 17789:2014, 3.2.7]

3.13 **поток** (stream): Упорядоченная последовательность передаваемых объектов, прикрепленных к порту принимаемых объектов.

[ИСО/МЭК 10179:1996, 4.33, изменено — путем удаления начального артикля и точки в конце]

3.14 **подроль** (sub-role): Подмножество деятельностей (3.4) данной роли (3.12).

[ИСО/МЭК 17789:2014, 3.2.9]

3.15 **управление данными** (data management): Совокупность деятельностей (3.4), направленных на имплементацию архитектуры больших данных, которая наиболее отвечает бизнес-целям в соответствии со стратегическим планом оценки управления данными.

3.16 **жизненный цикл данных** (data lifecycle): Стадии управления данными.

Примечание 1 — Цель жизненного цикла (определенная в ИСО 55000) — это данные в настоящем стандарте.

3.17 **прикладной программный интерфейс**; API (application programming interface, API): Граница, через которую прикладное программное обеспечение использует средства языков программирования для вызова сервиса.

[ИСО/МЭК 18012-2:2012, 3.1.4, изменено — примечание 1 к записи удалено, а конечная часть удалена из определения]

4 Сокращения

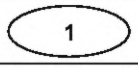
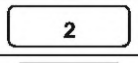
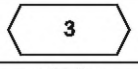
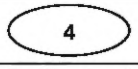
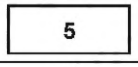
В настоящем стандарте применены следующие сокращения*:

ACID — атомарность, непротиворечивость, изоляция и долговечность (atomicity, consistency, isolation, and durability);
API — прикладной программный интерфейс (application programming interface);
BDA — аудитор больших данных (big data auditor);
BDAcP — сервис-провайдер доступа к большим данным (big data access provider);
BDAnP — сервис-провайдер аналитики больших данных (big data analytics provider);
BDAP — сервис-провайдер приложения больших данных (big data application provider);
BDC — потребитель больших данных (big data consumer);
BDCP — сервис-провайдер сбора коллекций больших данных (big data collection provider);
BDFP — сервис-провайдер среды обработки больших данных (big data framework provider);
BDIP — сервис-провайдер инфраструктуры больших данных (big data infrastructure provider);
BDP — сервис-провайдер больших данных (big data provider);
BDPlaP — сервис-провайдер платформы больших данных (big data platform provider);
BDPreP — сервис-провайдер предобработки больших данных (big data preparation provider);
BDProP — сервис-провайдер обработки больших данных (big data processing provider);
BDRA — эталонная архитектура больших данных (big data reference architecture);
BDSD — разработчик сервиса больших данных (big data service developer);
BDSO — оркестратор системы больших данных (big data system orchestrator);
BDSP — партнер сервиса больших данных (big data service partner);
BDVP — сервис-провайдер визуализации больших данных (big data visualization provider);
DG — стратегическое управление данными (data governance);
DM — менеджер данных (data manager);
DQM — менеджер качества данных (data quality manager);
PII — личная идентифицируемая информация (personally identifiable information);
RA — эталонная архитектура (reference architecture).

5 Условные обозначения

Схемы, представленные в настоящем стандарте, построены с использованием условных обозначений, показанных в таблице 1. Эти обозначения используют согласно описанию в ИСО/МЭК 17789.

Таблица 1 — Пояснения к диаграммам, используемым в настоящем стандарте

Объект	Обозначение
	Сторона
	Роль
	Подроль
	Деятельность
	Функциональный компонент
	Сквозной аспект

* *В список сокращений не включены сокращения CEP (complex event processing) и CPU (central processing unit), представленные в оригинале, т. к. в текстах оригинала и настоящего стандарта они не используются.*

6 Концепция эталонной архитектуры больших данных

6.1 Общие положения

В настоящем стандарте определена эталонная архитектура больших данных, которая служит фундаментальной точкой отсчета для стандартизации больших данных и представляет общую структуру архитектуры для описания основных концепций и принципов, лежащих в основе системы больших данных.

Настоящий стандарт описывает логические взаимосвязи (соотношения): между ролями/подролями, видами деятельности и функциональными компонентами, а также сквозные аспекты, составляющие архитектуру системы больших данных.

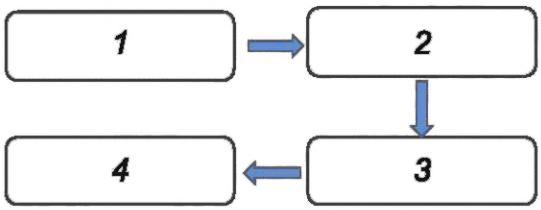
На рассмотренные взаимосвязи (отношения) распространены и некоторые другие стандарты, которые могут быть использованы:

- для уточнения уровня потока информации или других типов интероперабельности и/или
- для обеспечения заданных уровней качества (например, уровня защищенности или уровня сервиса).

Логические взаимосвязи, определенные в рамках эталонной архитектуры больших данных, являются существенной частью ее спецификации и функционирования и касаются таких вопросов, как категории информационных потоков между ее функциональными компонентами.

6.2 Представления

Большие данные могут быть описаны с применением представлений. В эталонной архитектуре больших данных использованы четыре вида различных представлений (см. рисунок 1 и таблицу 2).



1 — пользовательское представление; *2* — функциональное представление; *3* — представление реализации; *4* — представление развертывания

Рисунок 1 — Взаимосвязи между архитектурными представлениями

Таблица 2 — Представления в рамках эталонной архитектуры больших данных

Представление	Описание представления	Области применения
Пользовательское представление	Экосистема больших данных с заинтересованными сторонами (используется в ISO/IEC/IEEE 4210), роли, подроли и деятельность с большими данными	В рамках эталонной архитектуры больших данных
Функциональное представление	Функции, необходимые для поддержки деятельности с большими данными	В рамках эталонной архитектуры больших данных
Представление реализации	Функции, необходимые для реализации больших данных в сервисных компонентах и/или компонентах инфраструктуры	Вне рамок эталонной архитектуры больших данных
Представление развертывания	Иллюстрация того, каким образом технически реализованы функции больших данных в существующих компонентах инфраструктуры или в новых компонентах, которые будут встроены в эту инфраструктуру	Вне рамок эталонной архитектуры больших данных

Примечание — В настоящем стандарте рассмотрены детали пользовательского и функционального представлений, в то же время представления реализации и развертывания связаны с технологиями и реализа-

циями больших данных конкретных поставщиков и фактическими вариантами развертывания и, следовательно, выходят за рамки настоящего стандарта.

6.3 Обзор пользовательского представления

В рамках пользовательского представления в экосистеме больших данных использованы следующие понятия:

- стороны: сторона представляет собой физическое или юридическое лицо независимо от организационно-правовой формы или их группа; физические или юридические лица в экосистеме больших данных являются ее заинтересованными сторонами;

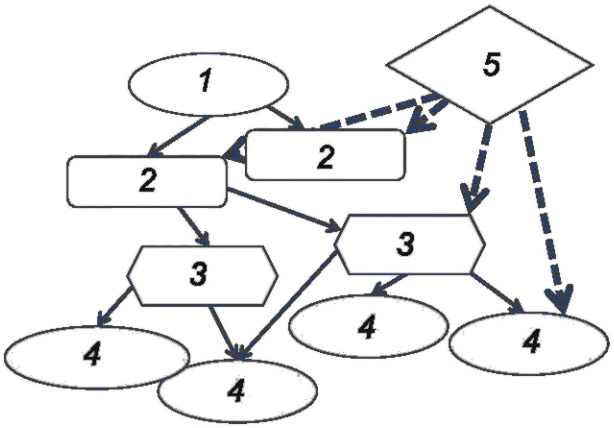
- роли и подроли: роль представляет собой набор деятельностей с большими данными, служащих общей цели: подроль — это подмножество деятельностей с большими данными для данной роли, и разные подроли могут совместно использовать деятельности, ассоциированные с данной ролью;

- виды деятельности: деятельность представляет собой определенное исполнение или набор задач, которые должны иметь цель и обеспечивать получение одного или нескольких результатов; деятельность реализуется с использованием функциональных компонентов;

- сквозные аспекты: сквозные аспекты могут быть общими и влиять на несколько ролей и на несколько видов деятельности с большими данными; сквозные аспекты сопоставляются с многоуровневыми функциями и связанными с ними функциональными компонентами, которые реализуют те или иные виды деятельности в рамках сквозного аспекта.

Примечание — Сторона может взять на себя более одной роли в любой конкретный момент и может участвовать в определенном подмножестве видов деятельности в рамках этой роли. Примеры сторон включают, помимо прочего, крупные корпорации, малые и средние предприятия, государственные организации, академические учреждения и частных лиц.

На рисунке 2 показаны взаимосвязи между сущностями пользовательского представления.



1 — сторона; *2* — роль; *3* — подроль; *4* — деятельность; *5* — сквозной аспект

Рисунок 2 — Сущности пользовательского представления

6.4 Обзор функционального представления

Функциональное представление является технологически нейтральным описанием функций, необходимых для формирования системы больших данных. Оно описывает распределение функций, необходимых для поддержания различных видов деятельности с большими данными. Функциональная архитектура также определяет взаимозависимости между функциями.

В архитектуре больших данных рассмотрены следующие понятия в рамках функционального представления:

- функциональные компоненты: функциональный компонент представляет собой такой функциональный строительный блок, который предназначен для участия в той или иной деятельности в процессе имплементации;

- функциональные уровни: уровень представляет собой набор функциональных компонентов со схожими возможностями или служащих общей цели;

- многоуровневые функции: многоуровневые функции включают в себя функциональные компоненты с возможностями, используемыми на нескольких функциональных уровнях; они также могут быть сгруппированы в подмножества.

Примечание — Создание функциональных компонентов в конкретной системе больших данных на всех уровнях является необязательным.

На рисунке 3 показаны сущности, определенные для пользовательского представления.

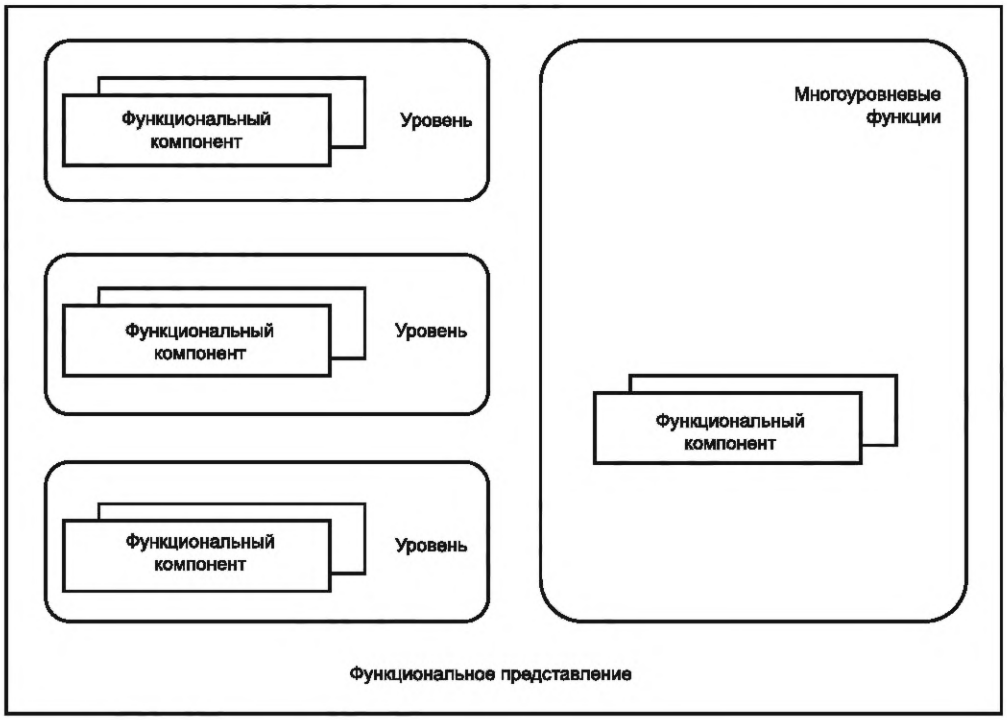


Рисунок 3 — Функциональные уровни

6.5 Взаимосвязи между пользовательским и функциональным представлениями

На рисунке 4 показана взаимосвязь пользовательского представления, отражающего совокупность видов деятельности с большими данными, и функционального представления в виде набора функциональных компонентов.

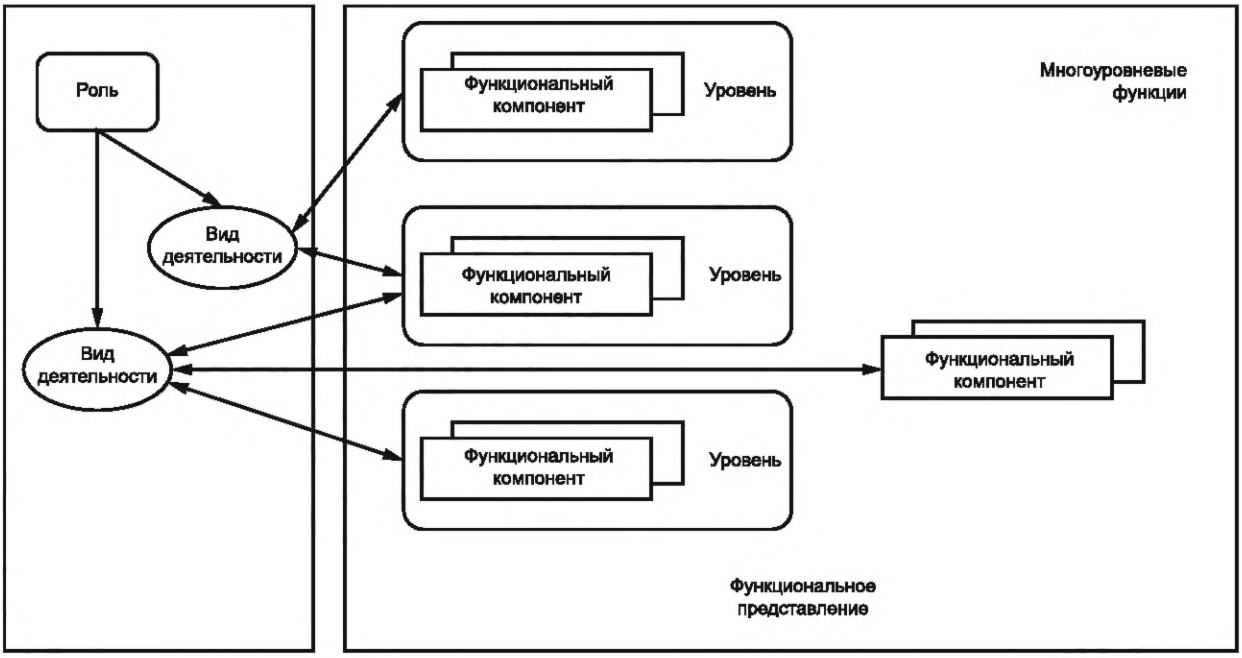


Рисунок 4 — Взаимосвязь пользовательского и функционального представлений

6.6 Взаимосвязи между пользовательским и функциональным представлениями со сквозными аспектами

Сквозные аспекты, как следует из их наименования, применимы как к пользовательскому представлению, так и к функциональному представлению больших данных.

В пользовательском представлении сквозные аспекты применяются к ролям и подролям и прямо или косвенно влияют на выполняемые этими ролями различные виды деятельности.

В функциональном представлении сквозные аспекты также относятся к функциональным компонентам, которые используют при выполнении видов деятельности, описанных в пользовательском представлении (см. рисунок 4).

Сквозные аспекты больших данных описаны в разделе 8 и включают:

- безопасность персональных данных;
- оперативное управление данными;
- стратегическое управление данными.

7 Пользовательское представление

7.1 Роли, подроли и виды деятельности с большими данными

Учитывая, что распределенные услуги и их предоставление лежат в основе применения больших данных, все виды деятельности, связанные с большими данными, можно разделить на три основные группы: виды деятельности, использующие большие данные; виды деятельности, предоставляющие услуги анализа больших данных; виды деятельности, предоставляющие данные.

В настоящем разделе представлены описания некоторых общих ролей и подролей, связанных с большими данными.

Необходимо отметить, что у стороны может быть более одной роли в любой момент времени. Однако роль, которую исполняет сторона, может ограничиться исполнением одной или нескольких подролей. Подроли — это подмножество видов деятельности в рамках конкретной роли с большими данными.

Как показано на рисунке 5, в архитектуре больших данных выделяют следующие роли:

- сервис-провайдер приложения больших данных (BDAP) (см. 7.2);
- сервис-провайдер среды обработки больших данных (BDFP) (см. 7.3);

- партнер сервиса больших данных (BDSP) (см. 7.4);
- сервис-провайдер больших данных (BDP) (см. 7.5);
- потребитель больших данных (BDC) (см. 7.6).

Примечание — Сервис-провайдер больших данных представляет собой любого поставщика данных для BDRA.

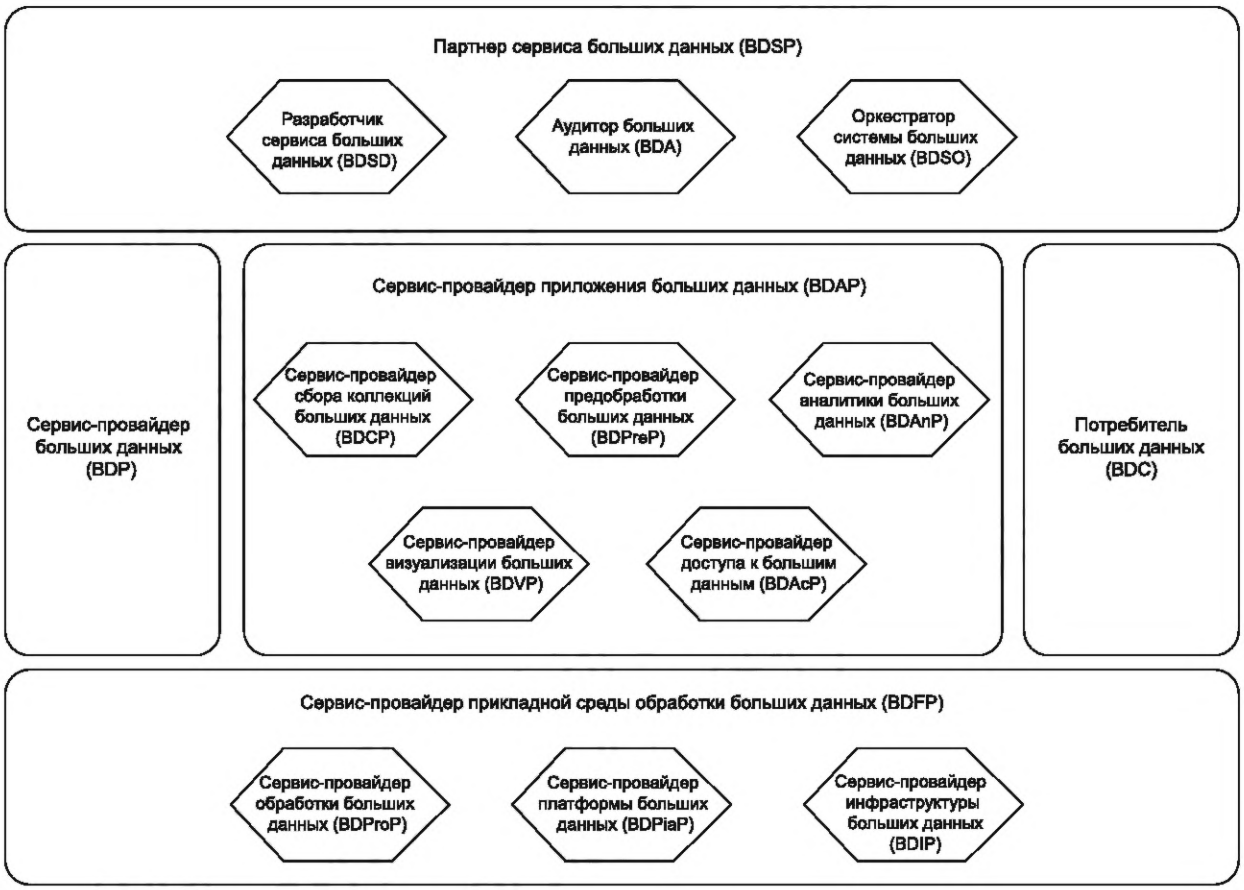


Рисунок 5 — Роли в архитектуре больших данных

В приложении В приведены примеры взаимосвязи ролей в экосистемах больших данных.

Каждая из подролей на рисунке 5 более подробно описана в 7.2—7.6.

7.2 Роль: сервис-провайдер приложения больших данных (BDAP)

7.2.1 Общие положения

Роль BDAP реализует свои функции на различных этапах жизненного цикла данных. В указанной роли объединены общие возможности пользовательского представления эталонной архитектуры больших данных, отраженные на рисунке 5 и представляющие комбинацию для построения конкретной системы данных.

Примечание 1 — Несмотря на то что действия роли BDAP одинаковы независимо от того, касается ли создаваемое решение больших данных или не касается, методы и приемы работы с большими данными изменяются в связи с тем, что данные и их обработка распараллеливаются между ресурсами.

Примечание 2 — Поскольку данные распространяются в рамках экосистемы, они обрабатываются и преобразуются для извлечения результатов из исходной информации различными способами. Каждый вид деятельности роли BDAP может быть реализован независимыми заинтересованными сторонами и представлен как автономный сервис.

Примечание 3 — Роль BDAP может быть отдельным экземпляром или совокупностью специализированных ролей BDAP, каждый(ая) из которых реализует различные этапы жизненного цикла данных. Каждый из видов деятельности роли BDAP может представлять собой общий сервис, вызываемый поставщиком или по-

требителем больших данных (например: веб-сервером, файловым сервером, набором из нескольких прикладных программ или их комбинации).

Примечание 4 — Роль BDAP обеспечивает реализацию, тестирование и валидацию бизнес-правил, а также требований и метрик качества данных, которые обеспечивают корректное управление данными в системе больших данных. Любая роль BDAP может обеспечить выполнение требований к качеству данных на протяжении всего жизненного цикла данных.

Роль BDAP включает следующие пять подролей, как показано на рисунке 6:

- сервис-провайдер сбора коллекций больших данных (см. 7.2.2);
- сервис-провайдер предобработки больших данных (см. 7.2.3);
- сервис-провайдер аналитики больших данных (см. 7.2.4);
- сервис-провайдер визуализации больших данных (см. 7.2.5);
- сервис-провайдер доступа к большим данным (см. 7.2.6).

Рисунок 6 — Виды деятельности подролей BDAP

7.2.2 Подроль: сервис-провайдер сбора коллекций больших данных

Подроль BDCP сервис-провайдера приложения больших данных обеспечивает сбор коллекций больших данных от провайдера данных. Таким сервис-провайдером может быть общий сервис, предоставляемый файловым сервером, или веб-сервером для приема или выполнения конкретных наборов данных, или специализированным сервисом, предназначенным для извлечения данных или получения данных в форме пуш-уведомлений от провайдера данных.

Подроль BDCP включает следующие виды деятельности:

- «найти источник данных» — направлена на поиск и хранение метаданных источников данных, которые можно использовать для сбора и/или хранения данных;
- «собрать данные» — направлена на преобразование доступных данных (например, веб-документ, данные блога и т. д.) в форму, которая может быть обработана системой;
- «регистрировать и буферизировать данные» — направлена на сохранение данных в реестре данных или временное хранение данных перед их передачей другим задачам или процессам.

7.2.3 Подроль: сервис-провайдер предобработки больших данных

Подроль BDPreP сервис-провайдера приложения больших данных обеспечивает подготовку необработанных данных к проведению анализа.

Подроль BDPreP включает следующие виды деятельности:

- «преобразовать данные» — направлена на преобразование данных или информации из одного формата в другой;
- «валидировать данные» — направлена на обеспечение корректности данных с использованием допустимых условий, таких как корректность, значимость, безопасность и конфиденциальность и т. д.;
- «очистить данные» — направлена на обнаружение некорректной части данных и их исправление путем замены, изменения или удаления;
- «агрегировать данные» — направлена на объединение двух или более наборов данных в один сводный набор данных.

Валидацию и очистку данных следует проводить с учетом требований системы менеджмента качества данных.

7.2.4 Подроль: сервис-провайдер аналитики больших данных

Подроль BDAnP сервис-провайдера приложения больших данных обеспечивает анализ больших данных в соответствии с требованиями, предъявляемыми к ним, для получения информации, которая соответствует технической цели.

Вид деятельности BDAnP представляет собой взаимосвязанное аналитическое логическое действие, которое включает моделирование процессов обработки данных в соответствии с логикой, предназначенной для извлечения информации из данных, на основе предъявляемых к приложению требований.

7.2.5 Подроль: сервис-провайдер визуализации больших данных

Подроль BDVP сервис-провайдера приложения больших данных обеспечивает представление потребителю больших данных информации об источнике данных или результатов анализа. Целью указанных видов деятельности является форматирование и представление данных таким образом, чтобы была обеспечена оптимальная передача как значения, так и смысла.

BDVP включает следующие виды деятельности:

- «опубликовать статус данных» — направлена на описание состояния данных в хранилище данных и включает различные виды визуализации, критерии классификации и т. д.;
- «форматировать результат анализа» — направлена на форматирование обработанных данных с целью надежной и эффективной коммуникации и включает визуальное представление, наложение и т. д.

7.2.6 Подроль: сервис-провайдер доступа к большим данным

Подроль BDAcP сервис-провайдера приложения больших данных обеспечивает обмен данными между приложением больших данных и провайдером больших данных или потребителем больших данных.

BDAcP включает вид деятельности «передать данные» (Transfer data), направленной на передачу или перезапись больших данных из одной системы в другую, обеспечивая их целостность, непрерывность и защищенность, а также конфиденциальность передачи данных.

7.3 Роль: сервис-провайдер среды обработки больших данных

7.3.1 Общие положения

Роль BDFP состоит из одного или нескольких иерархически организованных экземпляров компонентов архитектуры больших данных. При этом не требуется, чтобы все экземпляры на данном уровне иерархии реализовали одну и ту же технологию.

Примечание — На практике большинство реализаций больших данных представляют собой гибридные системы, сочетающие различные технологические подходы для обеспечения гибкости или удовлетворения всего спектра требований сервис-провайдера приложения больших данных.

Роль BDFP включает следующие три подроли (см. рисунок 7):

- сервис-провайдер инфраструктуры больших данных (см. 7.3.2);
- сервис-провайдер платформы больших данных (см. 7.3.3);
- сервис-провайдер обработки больших данных (см. 7.3.4).

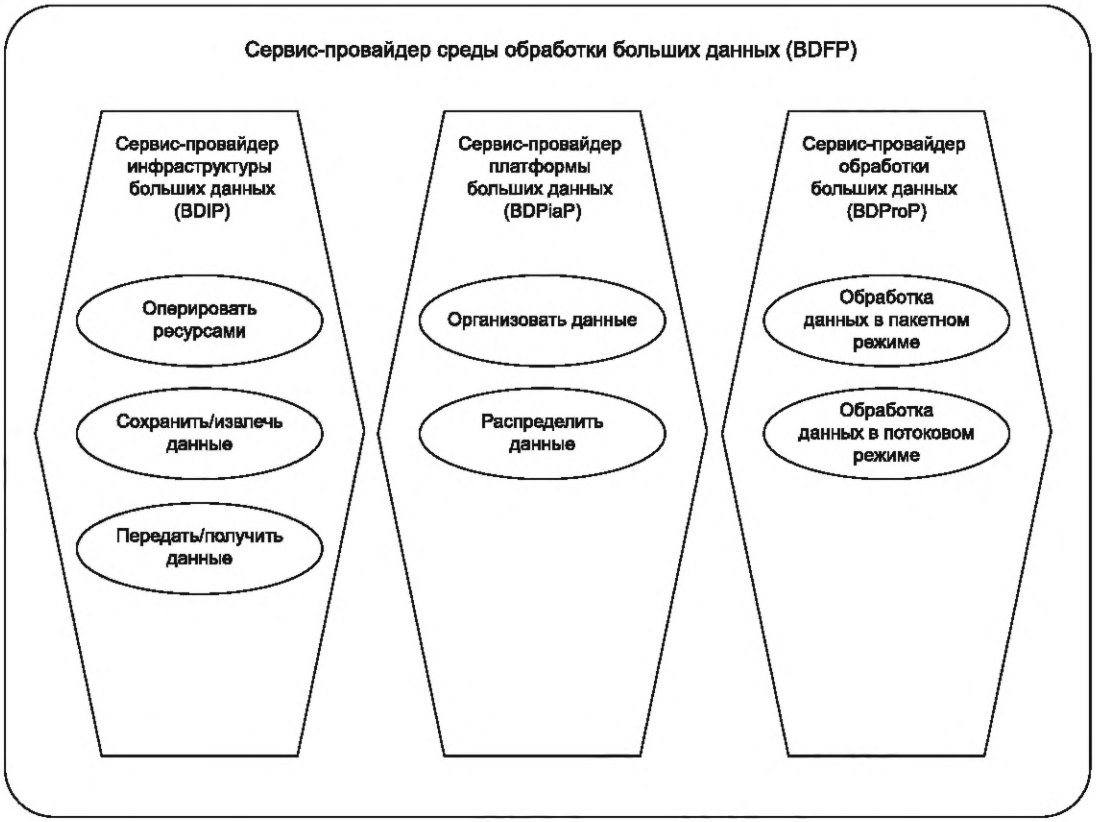


Рисунок 7 — Виды деятельности подролей сервис-провайдера среды обработки больших данных

7.3.2 Подроль: сервис-провайдер инфраструктуры больших данных

Подроль BDIP больших данных сервис-провайдера среды обработки больших данных обеспечивает предоставление системных ресурсов, включая системные средства (например, сети, компьютеры, хранилища и т. д.) и физическую среду (например, компьютерные залы, электропитание, кондиционеры и т. д.).

Подроль BDIP включает следующие виды деятельности:

- «оперировать ресурсами» — направлена на обработку или контроль физических или виртуальных ресурсов;
- «сохранить/извлечь данные» — направлена на сохранение и извлечение данных из хранилища;
- «передать/получить данные» — направлена на передачу данных по сети и доставку получателю.

7.3.3 Подроль: сервис-провайдер платформы больших данных

Подроль BDPLaP сервис-провайдера среды обработки больших данных, которая обеспечивает предоставление платформ для организации и распределения больших данных в рамках инфраструктуры больших данных и включает следующие виды деятельности:

- «организовать данные» — направлена на ранжирование, индексацию и связывание данных способами, предназначенными для конкретных приложений и аналитики;
- «распределить данные» — направлена на распределение данных между ресурсами инфраструктуры, чтобы максимально локализовать данные для обеспечения необходимого уровня производительности распределенных вычислений.

7.3.4 Подроль: сервис-провайдер обработки больших данных

Подроль BDProP сервис-провайдера среды обработки больших данных обеспечивает поддержку вычислительных и аналитических процессов для видов деятельности, выполняемых сервис-провайдером приложения больших данных.

Подроль BDProP включает следующие виды деятельности:

- «обработка данных в пакетном режиме» — обработка больших порций данных без обеспечения непрерывности обработки. Пакетная обработка используется, когда время отклика не является критичным. Пакетная обработка чаще всего связана с объемом данных или сложностью анализа;

- «обработка данных в потоковом режиме» — непрерывная обработка небольших порций данных (как правило, отдельные записи или элементы данных). Потоковая обработка используется, когда время отклика является критичным, и чаще всего связана со скоростью поступления данных.

7.4 Роль: партнер сервиса больших данных

7.4.1 Общие положения

Роль BDSP обеспечивает поддерживающий или вспомогательный вид деятельности между провайдером приложения больших данных, провайдером среды обработки больших данных, провайдером больших данных, потребителем больших данных или всеми перечисленными ролями.

В экосистеме больших данных виды деятельности роли BDSP в области больших данных варьируются в зависимости от типа партнера и его отношений с другими партнерами, с их ролями.

Роль BDSP включает следующие три подроли, как показано на рисунке 8:

- разработчик сервиса больших данных (см. 7.4.2);
- аудитор больших данных (см. 7.4.3);
- оркестратор системы больших данных (см. 7.4.4).

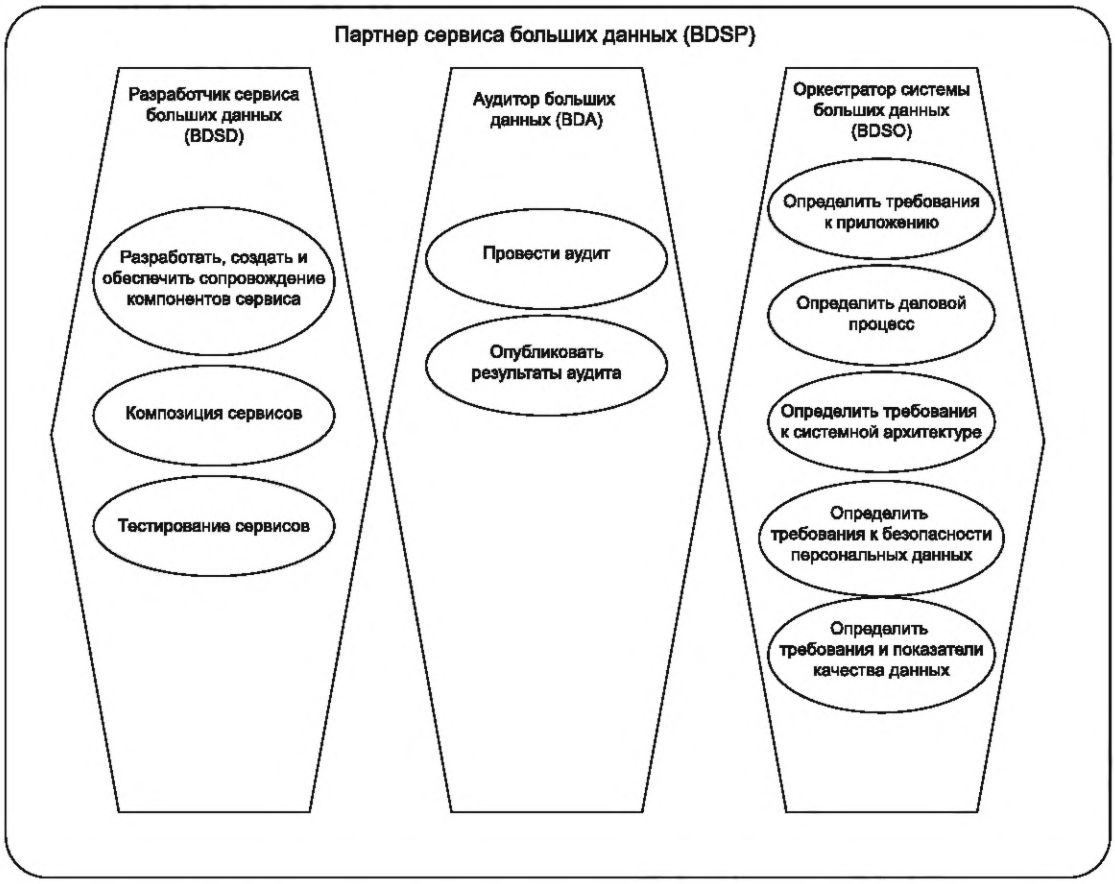


Рисунок 8 — Виды деятельности с большими данными, относящиеся к подролям партнера сервиса больших данных

7.4.2 Подроль: разработчик сервиса больших данных

Подроль BDSD партнера сервиса больших данных обеспечивает проектирование, разработку, тестирование и поддержку реализации сервиса больших данных, а также сопровождение процесса реализации сервиса больших данных.

Подроль BDSD включает следующие виды деятельности:

- «разработать, создать и обеспечить сопровождение компонентов сервиса» — направлена на разработку и создание программных компонентов, которые являются частью процесса реализации

сервиса больших данных, а также на предоставление рекомендаций по совершенствованию этого процесса;

- «композиция сервисов» — направлена на композицию сервисов с использованием существующих сервисов путем реализации функций посредничества и их агрегации;
- «тестирование сервисов» — направлена на тестирование компонентов и сервисов, созданных разработчиком сервиса больших данных.

7.4.3 Подроль: аудитор больших данных

Подроль BDA партнера сервиса больших данных обеспечивает проведение аудита вопросов предоставления и использования сервисов больших данных.

Подроль BDA охватывает такие вопросы, как достоверность источников данных, выполняемые операции, производительность, защищенность и конфиденциальность персональных данных, а также проверяет соответствие заданному набору критериев аудита.

Примечание 1 — Существуют различные спецификации для критериев аудита, например в [8] рассмотрены вопросы безопасности.

Подроль BDA включает следующие виды деятельности:

- «провести аудит» — направлена на запрос или получение результатов аудита, проведение необходимого тестирования системы или данных, подлежащих аудиту, а также получение результатов аудита программным путем;
- «опубликовать результаты аудита» — направлена на предоставление документированного отчета о результатах проведения аудита.

Примечание 2 — Подроль BDA обеспечивает оценку качества данных, определение и оценку уровней сервиса качества данных, непрерывное измерение и контроль качества данных.

7.4.4 Подроль: оркестратор системы больших данных

Подроль BDSO партнера сервиса больших данных обеспечивает описание основных требований к системе, включая требования к политике, стратегическому управлению, архитектуре, ресурсам, а также требований бизнеса и мониторинг видов деятельности для обеспечения соответствия системы этим требованиям.

Подроль BDSO включает следующие виды деятельности:

- «определить требования к приложению» — направлена на определение общих требований, которым должно удовлетворять приложение больших данных;
- «определить деловой процесс» — направлена на определение частично упорядоченного набора видов деятельности в рамках проекта, которые могут быть выполнены для реализации определенной цели проекта или части проекта и достижения некоторого ожидаемого конечного результата;
- «определить требования к системной архитектуре» — направлена на определение концептуальных требований к структуре, функционированию и облику системы больших данных;
- «определить требования к безопасности персональных данных» — направлена на определение требований к защищенности и конфиденциальности персональных данных с точки зрения стратегического управления;
- «определить требования и показатели качества данных» — направлена на повышение осведомленности о качестве данных, а также на определение бизнес-правил, требований и метрик качества данных.

7.5 Роль: сервис-провайдер больших данных

Роль BDP позволяет сделать данные доступными как для себя, так и для других. Выполняя свою роль, BDP позволяет формировать абстрактное представление различных типов источников данных, таких как необработанные данные или данные, ранее преобразованные другой системой, и позволяет делать их доступными через различные функциональные интерфейсы.

Примечание 1 — Концепция BDP не является новой, поскольку более широкие возможности сбора и анализа данных открывают новые возможности для предоставления значимых данных.

Роль BDP включает следующие виды деятельности (см. рисунок 9):

- «сделать данные доступными» — направлена на открытие или распространение источника данных за пределы первоначально предназначенной системы;

- «абстрагировать тип источника данных» — направлена на публикацию метаданных или каталога данных с целью распространения данных через реестр.

Рисунок 9 — Виды деятельности с большими данными, связанные с сервис-провайдером больших данных

Примечание 2 — При предоставлении данных другим лицам роль BDP может обеспечивать отслеживание данных и управление выявленными проблемами качества данных в соответствии с требованиями по управлению качеством данных.

7.6 Роль: потребитель больших данных

Роль BDC обеспечивает получение результатов работы системы больших данных.

Во многих отношениях она является получателем тех же функциональных интерфейсов, которые сервис-провайдер больших данных предоставляет сервис-провайдеру приложений для больших данных. После того как системой больших данных будут определены значения исходных данных, сервис-провайдер приложения больших данных в последующем обеспечивает предоставление потребителю больших данных функциональных интерфейсов соответствующего типа.

Роль BDC включает следующие виды деятельности (см. рисунок 10):

- «использовать большие данные» — направлена на использование результатов анализа больших данных или использование интерфейсов приложения, предоставляемых провайдером приложения больших данных для деловых целей потребителя больших данных;
- «оценить большие данные» — направлена на оценку качества больших данных или приложений больших данных в качестве обратной связи.

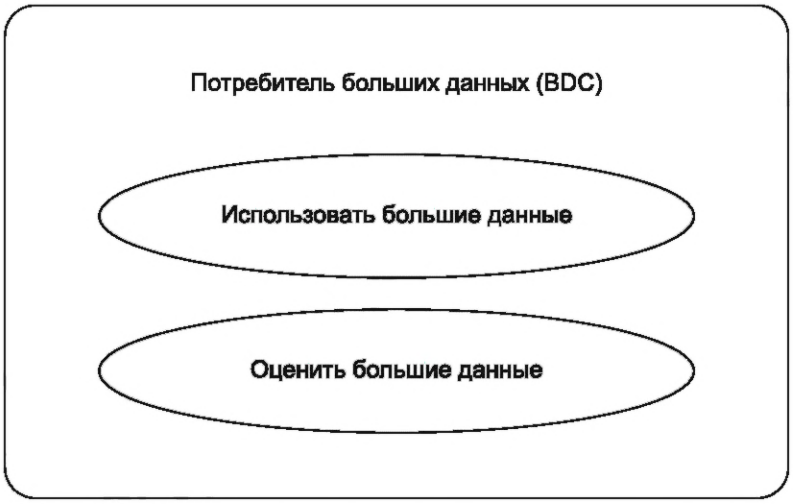


Рисунок 10 — Виды деятельности с большими данными, связанные с потребителем больших данных

8 Сквозные аспекты

8.1 Общие положения

К сквозным аспектам относят:

- безопасность больших данных и конфиденциальность персональных данных — указанный аспект касается того, как системы и данные защищены от риска путем сохранения их конфиденциальности, целостности и доступности, а также того, как персональные данные защищены от несанкционированного использования;
- оперативное управление данными — указанный аспект относится к тому, как системные компоненты и ресурсы выделяются, настраиваются, используются и контролируются;
- стратегическое управление данными — указанный аспект относится к тому, как данные контролируются и управляются в системе на протяжении этапов их жизненного цикла.

8.2 Безопасность больших данных и конфиденциальность персональных данных

Вопросы безопасности больших данных и конфиденциальности персональных данных затрагивают все остальные роли и подроли в экосистеме больших данных и функциональных компонентах эталонной архитектуры больших данных. Безопасность и конфиденциальность персональных данных тесно связаны с оркестратором системы больших данных в части политики, требований и проведения аудита, а также сервис-провайдером среды обработки больших данных и сервис-провайдером инфраструктуры больших данных в части разработки, развертывания и эксплуатации системы.

Проблематика обеспечения безопасности больших данных включает:

- конфиденциальность, обеспечивающую недоступность систем или данных неавторизованным лицам, организациям или процессам;
- целостность, обеспечивающую точность и полноту систем и данных;
- доступность, обеспечивающую доступность систем и данных и возможность их использования авторизованным органом по требованию.

Проблематика обеспечения конфиденциальности персональных данных в системах больших данных включает:

- несвязываемость, обеспечивающую гарантию того, что субъект персональных данных может многократно использовать ресурсы или услуги, при этом другие субъекты не смогут связать эти виды использования вместе;
- прозрачность, обеспечивающую достижение надлежащего уровня ясности процессов обработки данных, имеющих отношение к персональным данным, с тем чтобы сбор, обработка и использование информации могли быть понятны и восстановлены в любое время;

- возможность легитимного вмешательства, обеспечивающую гарантию того, что субъекты персональных данных, операторы персональных данных, обработчики персональных данных, а также надзорные органы могут вмешиваться во все процессы обработки данных, связанные с персональными данными [28], [29].

8.3 Оперативное управление данными

Такие характеристики больших данных, как объем, скорость, разнообразие и изменчивость, требуют универсальной платформы управления системой и программным обеспечением для решения задач предоставления, настройки и управления программным обеспечением и пакетной обработкой, мониторинга производительности, а также управления ресурсами и производительностью. Оперативное управление большими данными включает решение задач, связанных с системой больших данных, непосредственно данными, безопасностью данных и конфиденциальностью персональных данных с учетом их масштабирования при сохранении высокого уровня качества данных и безопасного доступа.

Проблематика обеспечения оперативного управления большими данными включает следующие вопросы:

- выделение ресурсов — деятельность по конфигурированию системных ресурсов для поддержки решения конкретной задачи; может выполняться на нескольких уровнях архитектуры системы от выделения ресурсов для виртуальных машин до выделения ресурсов для конкретного задания на одном или нескольких узлах; указанные мероприятия включают эффективное использование и конфигурирование ресурсов для поддержки решения одной или нескольких задач;
- конфигурация — обеспечивает надлежащую настройку параметров внутри системных компонентов для достижения оптимального функционирования и использования системных ресурсов;
- управление пакетами — обеспечивает управление базовыми наборами пакетов для системных компонентов с целью достижения требуемой безопасности и эксплуатационной надежности системы;
- управление ресурсами — обеспечивает использование ресурсов системы для поддержки различных рабочих нагрузок с учетом их приоритета.

8.4 Стратегическое управление данными

Стратегическое управление данными представляет собой высокоуровневый процесс планирования и реализации деятельности исполнителей различных ролей и подролей, направленный на создание новых ценностей и эффективное реагирование на потребности организации.

Стратегическое управление данными определяется и обеспечивается путем формирования:

- стратегии организации, связанной с управлением данными и обеспечивающей гарантию того, что данные соответствуют деятельности организации;
- стратегии управления качеством данных, представляющей собой набор ограничений, действий и требований, направленных на соответствие данных показателям качества, определяемых потребностями организации (более подробная информация представлена в приложении С).

9 Функциональное представление

9.1 Функциональная архитектура

9.1.1 Общие положения

Функциональная архитектура для больших данных позволяет описать большие данные в терминах высокоуровневого набора уровней функциональных компонентов. Функциональные уровни представляют собой наборы функциональных компонентов с аналогичными возможностями, которые требуются для выполнения различных видов деятельности с большими данными, представленных в разделе 8, для различных ролей и подролей применительно к большим данным и с учетом спецификации и реализации архитектуры больших данных.

Функциональная архитектура описывает функциональные компоненты с точки зрения многоуровневой архитектуры, в которой определенные типы функций сгруппированы на каждом уровне, как показано на рисунке 12.

Как показано на рисунке 11, роли и виды деятельности пользовательского представления больших данных, включая сервис-провайдера больших данных, потребителя больших данных, партнера сервиса больших данных, сервис-провайдера приложения больших данных и сервис-провайдера

среды обработки больших данных, реализуются с помощью четырехуровневых функций и/или многоуровневых функций.

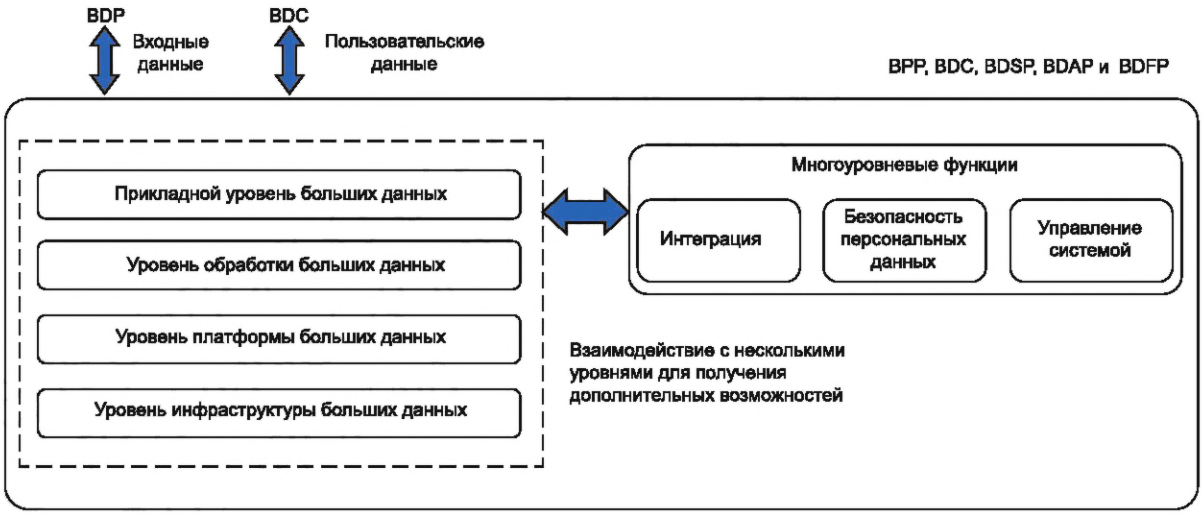


Рисунок 11 — Функциональные уровни эталонной архитектуры больших данных

Как показано выше, BDP и BDC могут быть внешними компонентами по отношению к системе больших данных в процессе разработки архитектуры или внутренними компонентами (поскольку один поставщик приложений в архитектуре больших данных может предоставлять входные данные или получать входные данные от другого поставщика приложений в рамках архитектуры). Дополнительная информация о сопоставлении функционального представления эталонной архитектуры больших данных с другой эталонной архитектурой системной интеграции представлена в приложении А.

В целях определения конкретной архитектуры рекомендуется использовать передовой опыт, чтобы архитектор задокументировал конкретные функциональные компоненты, обеспечивающие интерфейсы между этими уровнями и архитектурой больших данных.

9.1.2 Многоуровневая архитектура

9.1.2.1 Общие положения

Многоуровневая эталонная архитектура больших данных включает четыре уровня и набор функций, охватывающих все уровни.

Указанные четыре функциональных уровня включают:

- прикладной уровень больших данных (см. 9.1.2.2);
- уровень обработки больших данных (см. 9.1.2.3);
- уровень платформы больших данных (см. 9.1.2.4);
- уровень инфраструктуры больших данных (см. 9.1.2.5).

Функции, охватывающие различные уровни, называются многоуровневыми функциями.

На рисунке 11 представлена многоуровневая архитектура, при этом каждый из внутренних уровней многоуровневой архитектуры более подробно описан в 9.1.2.2—9.1.2.5.

9.1.2.2 Прикладной уровень больших данных

Прикладной уровень больших данных предоставляет функции поддержки приложения, включая сбор данных, подготовку, аналитику, визуализацию и функции доступа. Эти функции реализуются через интерфейсы сервис-провайдером больших данных на уровне обработки больших данных и уровне платформы больших данных, а также потребителем больших данных.

9.1.2.3 Уровень обработки больших данных

Уровень обработки больших данных предоставляет компоненты платформы и библиотеки для реализации аналитики, заданной уровнем провайдера приложений. На этом уровне его компоненты управляют выполнением аналитических задач в системе. Компоненты взаимодействуют с уровнем платформы с целью определения места, в котором хранятся данные в системе, и направляют результаты аналитики этих данных на соответствующий узел для того, чтобы обеспечить локализацию данных для выполнения вычислений. В рамках многоуровневых функций для обеспечения балансировки вычислений в системе они также взаимодействуют с компонентами управления ресурсами.

9.1.2.4 Уровень платформы больших данных

Уровень платформы больших данных предоставляет компоненты для хранения и организации данных, обрабатываемых системой. Указанные компоненты используют ресурсы одноименного уровня и в случае применения оперативной памяти координируют необходимые ресурсы с компонентами управления ресурсами в многоуровневых функциях с учетом требований к ним. Компоненты уровня платформы предназначены в первую очередь для обеспечения эффективной организации данных для обеспечения доступа от провайдера приложений и уровней обработки внутри системы.

9.1.2.5 Уровень инфраструктуры больших данных

Уровень инфраструктуры больших данных охватывает те ресурсы, к которым относится оборудование, обычно используемое в центре обработки данных, такое как серверы, сетевые коммутаторы и маршрутизаторы, устройства хранения, а также соответствующее программное обеспечение, не связанное с большими данными и работающее на серверах и другом оборудовании, таком как хост-операционные системы, гипервизоры, драйверы устройств, в том числе системное программное обеспечение.

На уровне инфраструктуры больших данных также реализуются функциональные возможности сети передачи больших данных, которые обеспечивают базовую взаимосвязь между сервис-провайдером приложений больших данных и сервис-провайдером больших данных/потребителем больших данных, а также в рамках провайдера приложений для больших данных — между одноранговыми провайдерами приложений больших данных.

9.1.3 Многоуровневые функции

Многоуровневые функции включают ряд функциональных компонентов, которые взаимодействуют с функциональными компонентами вышеупомянутых четырех других уровней и обеспечивают, помимо всего прочего, следующие вспомогательные возможности:

- возможности функционирования систем безопасности (аутентификации, авторизации, аудита, валидации, шифрования);
- возможности интеграции (взаимодействия различных компонентов для достижения требуемой функциональности);
- возможности управления [развертыванием, конфигурацией, мониторингом, множественной арендой (мультитенантностью) ресурсов, высокой доступностью и жизненным циклом больших данных].

Многоуровневые функции, описанные выше, могут поддерживать сквозные аспекты или виды деятельности ролей, которые имеют широкое применение в системной архитектуре.

9.2 Функциональные компоненты

9.2.1 Общие положения

В данном подразделе архитектура больших данных описывается с точки зрения общего набора функциональных компонентов больших данных. Функциональный компонент является функциональным элементом эталонной архитектуры больших данных, который используется для выполнения видов деятельности или некоторой части вида деятельности и имеет артефакт выполнения в конкретном варианте архитектуры, например программный компонент, подсистему или приложение.

На рисунке 12 представлен общий вид функциональных компонентов эталонной архитектуры больших данных, организованных посредством многоуровневой архитектуры.

Термин «структура», используемый для имен функциональных компонентов на рисунке 12 и связанных с ним текстовых разделов, определен в ISO/IEEE 11073-10201 как «структура процессов и спецификаций, предназначенных для поддержки выполнения конкретной задачи».

Примечание — Учитывая диапазон приложений/областей, связанных с большими данными, и быстрое развитие технологий больших данных, описание исчерпывающего списка возможных функциональных компонентов на этих уровнях является объемным и не может быть полным. Поэтому в настоящем стандарте представлено только общее описание функциональных компонентов.

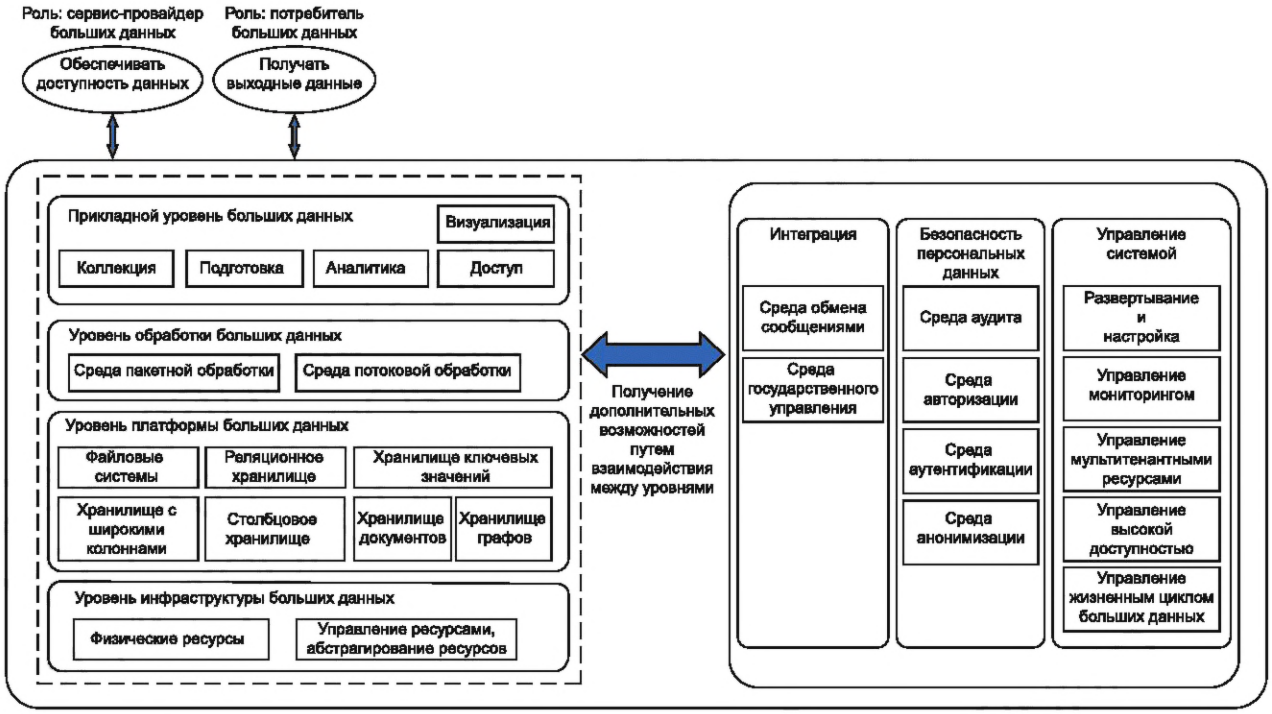


Рисунок 12 — Функциональные компоненты эталонной архитектуры больших данных

9.2.2 Функциональные компоненты прикладного уровня больших данных

9.2.2.1 Общие положения

Уровень приложений больших данных с функциональными компонентами поддерживает деятельность сервис-провайдера приложений больших данных и обеспечивает основной интерфейс для внешних компонентов, включая провайдеров и потребителей больших данных. В качестве компонентов в данном случае выступают компоненты на уровне обработки больших данных и на уровне платформы больших данных для реализации различных видов деятельности на уровне приложений для больших данных. Ниже приведены основные функциональные компоненты этого уровня.

9.2.2.2 Функциональные компоненты комплектования больших данных

Функциональные компоненты комплектования больших данных используются для создания механизмов импорта данных от провайдера данных, а также системы хранения данных для выполнения последующих процессов:

- установление связи;
- импорт данных;
- хранение данных.

Данная категория компонентов связана с получением данных в системе. Указанные компоненты могут эффективно реализовать свои функции с учетом объема и скорости поступающих данных.

9.2.2.3 Функциональные компоненты предобработки

Функциональный компонент предобработки используют для подготовки данных, предназначенных для конкретного процесса анализа. Детализированные функции включают: агрегацию данных, очистку данных, конверсию/трансформацию данных, создание вычислительного поля данных, оптимизацию данных, разделение данных, суммирование данных, выравнивание данных, проверку данных, виртуализацию и хранение подготовленных данных. Виртуализация данных представляет собой подход к управлению данными, при котором приложение может получать доступ к данным и изменять их, не выполняя физического форматирования и хранения данных. Трансформация данных преобразует данные из одного формата в другой, включая шифрование/дешифрование, компрессию/декомпрессию, прореживание, получение сводных данных и нормализацию данных.

9.2.2.4 Функциональный компонент аналитики

Функциональный компонент аналитики используют для инкапсуляции специализированных вычислений, которые необходимо выполнять с данными для поиска информации и/или извлечения знаний для удовлетворения прикладных требований с применением заданных алгоритмов.

Примечание 1 — Классы алгоритмов для машинного обучения реализуют в том числе функции: корреляции, классификации, слияния данных, интеграции данных, интеллектуального анализа данных, искусственного интеллекта, распознавания образов, прогнозного моделирования, регрессии, кластерного анализа, пространственного анализа, аудиоанализа, визуального анализа, текстового анализа и др. К алгоритмам текстового анализа относятся методы: анализа тональности, распознавания именованных объектов и определения темы; к алгоритмам машинного обучения относятся методы: корреляции, классификации, распознавания образов, прогнозного моделирования, регрессии, кластерного анализа и пространственного анализа. Во многих случаях системы больших данных объединяют несколько таких типов алгоритмов в потоковый процесс обработки данных. Например, система может использовать распознавание именованных сущностей для извлечения определенных сущностей (людей, мест, организаций и т. д.) из неструктурированных фрагментов текста, а затем передавать эту информацию в виде признаков для кластеризации фрагментов текста с использованием алгоритмов кластеризации на основе K-ближайших соседей или K-средних.

Примечание 2 — Классом аналитических функций является оперативный анализ данных, т. е. анализ лог-файлов, данных о системном статусе, предупредительной информации и др., предназначенных для эксплуатации и обслуживания системы. Типовой запрос и анализ включают: поиск текстового лог-файла, многомерный комплексный анализ и т. д. К алгоритмам численного анализа относятся: использование быстрого преобразования Фурье, линейной алгебры и методы *N*-тел. К графовым алгоритмам относятся: выявление массивов данных, поиск подграфа/смысла, оценку размерности поиска, коэффициента кластеризации, рейтинга страницы, максимальных кликов, компонентов связности, промежуточной центральности, кратчайшего пути.

Примечание 3 — Критические характеристики рассмотренных алгоритмов для больших данных определяются тем, что они должны иметь возможность работать параллельно на уровне обработки данных и учитывать распределенный характер данных на уровне платформы.

9.2.2.5 Функциональный компонент визуализации

Функциональный компонент визуализации используют для взвешенного представления проанализированных данных потребителю больших данных. Детализация задач функционального компонента включает визуализацию:

- результатов разведочного анализа данных (многомерность, переменная разрешающая способность, взаимодействие, анимация, симуляция, статистическая графика, рендеринг поверхности, рендеринг объема);
- знаниевого компонента/объяснительного компонента (отчеты, бизнес-аналитика и обобщающая презентация для клиентов).

Примечание — Значимым аспектом визуализации больших данных является представление больших наборов данных таким образом, чтобы по ним можно было четко ориентироваться и они были доступными для понимания. Кроме того, может потребоваться работа с данными в распределенном параллельном режиме.

9.2.2.6 Функциональный компонент доступа

Функциональный компонент доступа используют для предоставления потребителям больших данных доступа к результатам прикладного уровня больших данных. Детализированные функции включают:

- управление правами доступа;
- экспорт данных (например, через программный интерфейс приложения, протокол или язык запросов);
- безопасный доступ к данным.

Примечание — Потребители больших данных подключаются через указанный функциональный компонент с помощью веб-сервисов, пользовательских интерфейсов и/или API, протоколов и т. д., применяемых для доступа/извлечения данных. Уникальная задача для больших данных состоит в сложности предоставления потребителю больших данных доступа к ним с учетом их объемов и скорости обработки.

9.2.3 Функциональные компоненты уровня обработки

9.2.3.1 Общие положения

Компоненты уровня обработки больших данных в первую очередь ориентированы на показатели производительности (например, получение результатов вычислений за требуемый период времени). Уровень обработки больших данных предоставляет функциональные компоненты для поддержки таких характеристик больших данных, как объем, скорость обработки и разнообразие. Указанный уровень использует различные механизмы обработки для различных хранилищ данных и планирования вычислений в ближнем или локальном хранилище. Он обеспечивает абстрактную функциональность для выполнения операций прикладного уровня больших данных. Пользовательские операции абстрагируются в качестве: источника данных, фильтра, карты, окна, агрегации и др. Уровень обработки больших

данных завершает процесс обработки потока данных между операторами, а также входом и выходом. На этом уровне реализован процесс параллельной обработки данных.

Примечание 1 — В существующих системах баз данных компоненты уровня обработки больших данных носят наименование исполнительного механизма. Уровень обработки больших данных в большей степени относится к среде выполнения. Ключевое слово «большой» означает не только наличие больших данных из источника; при этом объем промежуточных данных может быть больше объема необработанных данных.

Примечание 2 — При распараллеливании операций компоненты уровня обработки обычно распределяют работу между узлами в кластере сначала на основе местоположения данных (например, данные на уровне платформы, необходимые для вычислений, находятся на узле), а затем на основе ресурсов памяти и ресурсов центрального процессора.

Примечание 3 — Примером этого является шаблон программирования map/reduce (отображение/свертка), в котором вычисления для отдельных записей распределены по узлам в зависимости от местоположения данных на этапе сопоставления, а затем результаты каждого узла объединяют и сортируют на этапе сокращения.

На уровне обработки больших данных использованы различные механизмы обработки для разных хранилищ данных и планирования вычислений в ближнем или локальном хранилище.

Как правило, платформы на уровне обработки больших данных классифицируют в зависимости от количества элементов и скорости их обработки. Распространенными формами оценки является один блок (пакет) или один элемент (поток).

9.2.3.2 Функциональный компонент среды пакетной обработки

Функциональный компонент пакетной обработки в основном направлен на решение задач обработки больших объемов данных. В качестве базовой единицы для пакетной обработки использована группа элементов. Полученные элементы блокируются для формирования пакета на основе их распределения на уровне платформы для обработки, чтобы максимизировать локальность данных. После того как каждый узел обработал очередной пакет, результаты синхронно или асинхронно пересылаются на следующий шаг, который может представлять другой цикл обработки (как это реализовано в массовом синхронном параллельном шаблоне) или суммирование результатов (как это реализовано в методе «отображение/шаблон свертки»). Время, необходимое для выполнения пакетной аналитики, может варьироваться от часов до долей секунды в зависимости от аналитики и данных. Приложениям специальных запросов и отчетов ежедневного оперативного анализа может потребоваться разное время отклика. [Когда время отклика находится в пределах минут, часов или дольше, это часто называют автономной обработкой. Если же время отклика составляет секунды и менее, это называется интерактивной обработкой.] Однако тот факт, что система спроектирована как интерактивная, не означает, что все время отклика находится в диапазоне секунд или долей секунды. Ненадлежащим образом выполненная(ый) аналитика (запрос), которая(ый) может иметь простые или сложные взаимосвязи между данными, должна (должен) обрабатывать большой объем записей, что может занять минуты или часы.

9.2.3.3 Функциональный компонент среды потоковой обработки

9.2.3.3.1 Общие положения

Функциональный компонент среды потоковой обработки в основном направлен на решение задачи обеспечения скорости обработки. Модель процесса определяется как конвейерная, и каждый элемент пересылается следующему оператору с минимальной задержкой. Главная задача заключается в получении мгновенного ответа, при этом каждый элемент определяется как значимый в данный момент времени, в то же время некоторые операции требуют, чтобы элементы были заблокированы или буферизованы, например в случае выполнения процесса агрегирования скользящего окна. Однако далее данные подвергают непрерывной конвейерной обработке. Функциональный компонент среды обмена сообщениями (см. 9.2.6.2.2) применяют для взаимосвязи между операторами через узлы. Когда данные имеют слишком большой объем и/или слишком высокую скорость, в системе больших данных может быть применено временное хранилище, удаление избыточных данных или использование совместно с производителем механизма ограничения скорости, чтобы избежать сбоев в системе.

Поток данных представляет основную характеристику потоковой среды и внутренне может быть описан ориентированным ациклическим графом, который в качестве вершины включает оператор, а в качестве ребер — поток событий. Оператор может быть распараллелен, а поток событий может быть разделен на порции. Обработка сложных событий определяется как более трудоемкая, чем обычная потоковая обработка, и к ней можно обращаться с запросами, что обеспечивает добавление дополнительных функциональных характеристик, таких как: упорядочение событий, гарантированная обработка событий, хранение состояний и разделение потока на порции/распараллеливание операций.

Указанные четыре характеристики описаны в 9.2.3.3.2—9.2.3.3.5.

9.2.3.3.2 Упорядочение событий

Упорядочение событий обеспечивается пользовательской глобальной меткой времени или идентификатором члена последовательности, оба из которых помечаются идентификатором одного потока. Упорядочение событий может осуществляться по времени наступления или количеству событий и зависит от оконного потока. Когда используется параметр времени события, упорядочение событий означает, что анализ события следует проводить в оконном операторе с учетом меток времени. Неупорядоченные и задержанные события необходимо переупорядочить, сбросить с обработки или немедленно проанализировать. Когда используется счетчик событий, упорядочение событий означает, что событие должно оцениваться в операторе окна посредством идентификатора члена последовательности. Время наступления события или идентификатор члена последовательности должны монотонно возрастать.

9.2.3.3.3 Гарантированная обработка событий

При наличии сбоев события должны быть обработаны с использованием отказоустойчивого механизма. Это имеет особенное значение в случае разделения потоковой передачи, параллельной работы оператора и наличия распределенности данных. Данные, хранящиеся в памяти, и данные, хранящиеся в файловой системе постоянного хранилища, должны быть гарантированно обработаны в оконном интервале. Пристальное внимание необходимо уделить двум существенным этапам: получение данных о событиях перед обработкой (приемником) и их фиксация после обработки (обработчиком).

Гарантированная обработка событий обычно включает три варианта:

- не более одного раза: такой вариант означает, что на этапе приема событие должно быть получено один раз от источника данных, при этом не должно поддерживаться полученное смещение, а этап функционирования обработчика не гарантируется. Полученное событие может быть обработано, при этом результаты обработки не возвращаются. Данный класс обработки является простым и имеет низкую задержку, при этом корректность обработки гарантируется не всегда;

- минимум один раз: такой вариант означает, что на этапе получения событие может быть воспроизведено и получено несколько раз, а на этапе функционирования обработчика события могут многократно обрабатываться. Все события могут быть получены и обработаны, но при этом результат может быть неточным. В этом случае необходимо поддерживать дополнительный механизм ручной корректировки систематического отклонения, позволяющий скорректировать последствия повторного воспроизведения событий; также механизм обработки дубликатов с целью сокращения времени повторной обработки одних и тех же событий. При этом возникают дополнительные накладные расходы, однако указанный механизм может обеспечить низкое значение задержки и определенную степень гарантированности обработки;

- однократно: событие принимается один раз и обрабатывается один раз без потерь и повторов. При этом гарантированы этапы работы приемника и обработчика. Оба этапа требуют реализации механизмов независимой отказоустойчивости и восстановления после сбоев для обеспечения работы неделимого и надежного хранилища. В этом случае возрастают накладные расходы из-за частых операций ввода-вывода, но при этом гарантируется корректность.

9.2.3.3.4 Хранилище состояний

Типовые потоковые среды обработки имеют модель конвейерного процесса, в то время как средам со сложной обработкой событий по сравнению с потоковыми средами требуется дополнительное состояние для поддержки операции окна. Операция окна предназначена для непрерывного запроса, когда событие сохраняется в течение определенного периода времени для создания окна. В традиционном варианте среды со сложной обработкой событий окно является маленьким, а событие сохраняется в буфере. В то время как события в окне в современной среде со сложной обработкой событий для больших данных могут быть множественными, хранилище состояний может обеспечить поддержку потоков с большим объемом событий. Дополнительное хранилище необходимо для поддержания отказоустойчивости и восстановления после сбоев, репликации, записи логов в журнал упреждающей записи и реализации контрольной точки, которые являются классическими методами решения указанных задач, поэтому хранилище состояний может поддерживать распределенную семантику и семантику ACID ограниченными способами, а компромисс заключается в обеспечении функций производительности и коррекции.

9.2.3.3.5 Разделение потока/распараллеливание операторов

Данная характеристика относится к вопросам масштабируемости. Функции потока и оператора выполняются в ориентированном ациклическом графе. Цель потоковых сред обработки данных — максимально распараллелить выполнение. Потоковое разделение служит для обеспечения распределения событий, а оператор parallel — для параллельных вычислений. Планировщик реализует параллельные

вычисления с локальными событиями. Разделение потока по ключу (например, идентификатор датчика, идентификатор пользователя, идентификатор учетной записи) и функция агрегирования оцениваются отдельно для разделенного потока. Потоковые метаданные, координация связи, динамическое распределение ресурсов и стратегия извлечения по принципу push/pull (активный и пассивный способы извлечения данных) необходимы для разделения потока в распределенной среде. Распараллеливание операторов необходимо для обеспечения быстрых вычислений, но для координации глобального состояния (барьер, порядок событий) требуется больше механизмов. Операторы достаточно часто выполняют свои функции последовательно друг за другом с целью уменьшения накладных расходов на сетевые коммуникации некоторые операторы могут быть объединены в цепочку.

9.2.4 Функциональные компоненты уровня платформы больших данных

9.2.4.1 Общие положения

Компоненты уровня платформы больших данных предоставляют услуги по хранению, организации и извлечению данных для поддержки более высоких уровней. Этот уровень обеспечивает логическую организацию и распределение данных в сочетании с соответствующими прикладными программными интерфейсами или методами доступа, а также реестром данных и сервисами оперирования метаданными, такими как семантические описания данных в виде формальных онтологий или таксономий.

Примечание — Одним из аспектов архитектуры этого уровня является выбор или совершенствование организации данных и методов хранения для более эффективного использования данных и повышения производительности запросов или процедур извлечения. В связи с быстрым увеличением объемов больших данных (например, в финансовой сфере, банковском деле, средствах массовой информации, обрабатывающей промышленности) и многообразием вариантов использования пользователям требуется повышенная производительность для реализации различных запросов и процедур анализа с меньшей степенью дублирования и избыточности в хранилище данных.

В 9.2.4.2—9.2.4.8 описаны общие категории этих компонентов.

9.2.4.2 Функциональный компонент файловой системы

Файловые системы организуют фрагменты данных (обычно определяемые как записи), доступ к которым осуществляется как к именованному объекту в определенном пространстве имен. В то время как локальные файловые системы часто применяют в системах больших данных для хранения промежуточных данных локально по отношению к узлу обработки, распределенные файловые системы гораздо более распространены для постоянного хранения данных. Разница заключается в том, что распределенные файловые системы обеспечивают управление распределением и репликацией блоков данных между узлами, при этом пространство имен не хранится вместе с данными, а управляется через центральную службу имен, часто работающую в режиме «ведущий/ведомый» или с несколькими ведущими для обеспечения отказоустойчивости.

Распределенные файловые системы (также известные как кластерные файловые системы) стремятся преодолеть проблемы с пропускной способностью, связанные с объемными и скоростными характеристиками больших данных, объединяя пропускную способность ввода-вывода для нескольких устройств (дисковых приводов) на каждом узле с избыточностью и отказоустойчивым зеркалированием или репликацией данных на уровне блоков между несколькими узлами. Процедура репликации данных распределенной файловой системы специально разработана для использования разнородного стандартного оборудования в кластере больших данных. Таким образом, в случае сбоя одного диска или всего узла данные не теряются, так как они реплицируются на другие узлы, а пропускная способность снижается лишь минимально, так как эта обработка может быть перенесена на другие узлы. Кроме того, репликация обеспечивает высокий уровень параллелизма при чтении данных и начальной записи.

Хранилища распределенных объектов (также известные как глобальные хранилища объектов) являются уникальным примером организации распределенной файловой системы. В отличие от подходов, описанных выше, которые реализуют традиционные подходы к пространству имен иерархии файловой системы, хранилища распределенных объектов представляют собой плоское пространство имен с глобальным уникальным идентификатором для любого фрагмента данных. Как правило, данные в хранилище находятся с помощью запроса к каталогу метаданных, который возвращает соответствующие идентификаторы. В общем случае с глобальными уникальными идентификаторами предоставляется базовая программная реализация с местом хранения данных. Такие хранилища разрабатываются и реализуются для хранения очень больших объектов данных — от полных наборов данных до крупных отдельных объектов (например, изображений с высоким разрешением размером в десятки гигабайт).

9.2.4.3 Функциональный компонент реляционного хранилища

В случае реляционной модели хранения данные хранятся в виде строк, где каждое поле представляет собой столбец, организованный в виде таблицы на основе логической организации данных.

Примечание — Реализации реляционных моделей хранения для больших данных являются относительно зрелыми и были приняты на вооружение рядом организаций. Реляционные модели также очень быстро совершенствуются благодаря новым реализациям, ориентированным на улучшение времени отклика. Многие реализации для больших данных используют подход «грубая сила» к масштабированию реляционных запросов. По существу запросы разделяют на этапы, но, что более важно, обработку входных таблиц распределяют между несколькими узлами (часто в виде задания сопоставления/уменьшения).

Фактическим хранилищем данных могут быть плоские файлы (с разделителями или фиксированной длины), где каждая запись/строка в файле представляет собой строку в таблице. Однако эти реализации все чаще используют двоичные форматы хранения, оптимизированные для распределенных файловых систем. В таких форматах часто применяются индексы на уровне блоков и организация данных по столбцам, чтобы обеспечить доступ к отдельным полям в записях без необходимости чтения всей записи. Несмотря на это большинство моделей реляционного хранения больших данных по-прежнему являются пакетно-ориентированными системами, предназначенными для сложных запросов, которые генерируют очень большие промежуточные матрицы перекрестных результатов из объединений, поэтому даже для выполнения простейшего запроса могут потребоваться десятки секунд.

9.2.4.4 Функциональный компонент хранилища «ключ — значение»

Принципы построения хранилищ «ключ — значение» лежат в основе всех других моделей хранения и индексации. С точки зрения больших данных эти хранилища эффективно представляют модели памяти с произвольным доступом. Хотя данные, хранящиеся в значениях, могут иметь достаточно сложную структуру, вся обработка этой структуры должна обеспечиваться приложением, а реализация хранилища часто возвращает только указатель на блок данных. Хранилища типа «ключ — значение» наиболее эффективны для отношений «1—1», один к одному (например, каждый ключ относится к одному значению), но также могут быть применены для сопоставления ключей со списками однородных значений. Когда ключи сопоставляют несколько значений разнородных типов/структур или когда значения из одного ключа необходимо объединить со значениями для другого или того же ключа, требуется пользовательская прикладная логика. Указанное требование для этой пользовательской логики часто препятствует решению задачи эффективного масштабирования.

В хранилищах типа «ключ — значение» обычно надлежащим образом реализованы обновления в случаях, когда имеет место отношение один к одному, а значение (размер/длина) данных не изменяется. Способность хранилищ «ключ — значение» обрабатывать вставки, как правило, зависит от базовой реализации. При этом хранилища типа «ключ — значение» требуют значительных ресурсов (как ручных, так и вычислительных) для обработки изменений в базовой структуре данных значений. Наиболее часто используемой реализацией в приложениях для работы с большими данными являются распределенные хранилища типа «ключ — значение». При этом обязательно должна быть решена одна задача, которая не может быть уникальной для реализаций типа «ключ — значение», — это распределение ключей по всему пространству возможных вариантов «ключ — значение».

В частности, ключи необходимо выбирать достаточно тщательно, чтобы избежать искажений в распределении данных по кластеру. Когда данные чрезмерно искажены в рамках небольшого диапазона, это может привести к возникновению горячих точек (hot spot) вычислений в кластере в том случае, если в процессе реализации предпринимается попытка оптимизировать локализацию данных. Если данные являются динамическими (добавляются новые ключи) для такой реализации, то вполне вероятно, что в какой-то момент данные могут потребовать повторной балансировки в кластере. В случае реализации, когда локализация не оптимизируется, используют различные виды хеширования, а также случайного или циклического подхода к распределению данных, как правило, не возникают искажения и горячие точки. Однако в этом случае особенно сложно решаются задачи, требующие агрегирования наборов данных.

9.2.4.5 Функциональный компонент столбцового хранилища с широкими столбцами

В отличие от простых реляционных данных, которые хранят данные по строкам связанных отношениями значений, в столбцовых хранилищах данные организуются в группы подобных значений. В этом случае разница является небольшой, но в реляционных базах данных отдельная группа столбцов (часто один или несколько столбцов) для создания записи связана с определенным первичным ключом.

В столбцовых хранилищах значение каждого столбца является ключом, и подобные значения столбцов указывают на связанные строки.

Простейший пример столбцового хранилища — это не более чем хранилище ключей и значений, в котором роли ключа и значения поменялись местами. Во многих отношениях столбцовые хранилища данных наиболее похожи на индексы в реляционных базах данных. Кроме того, реализация столбцовых хранилищ с широкими столбцами, которые соответствуют модели разреженной, распределенной многомерной модели отсортированной карты (где произвольные байтовые массивы индексируются/доступны на основе строковых и столбцовых ключей), вводят дополнительный уровень сегментации за пределами таблицы, строки и столбца реляционной модели, которая получила наименование семейства столбцов. Таким образом, в столбцовом хранилище с широкими столбцами добавляется дополнительное измерение, известное как семейство столбцов.

9.2.4.6 Функциональный компонент столбцового хранилища

Путем организации и хранения данных по столбцам (а не по строкам — в хранилищах на основе строк) столбцовые базы данных наиболее подходят для приложений с большими данными, которые требуют широкого спектра вариантов анализа, таких как многомерный OLAP (оперативная аналитическая обработка) — запрос, запрос сканирования (большого и малого). Для повышения производительности запросов могут быть применены различные методы сортировки, индексации и сжатия на основе столбцов, например: многомерное индексирование, словарное кодирование и т. д., используемые для повышения производительности запросов.

9.2.4.7 Функциональный компонент хранилища документов

В последнее время активно развивались и нашли широкое применение современные хранилища документов, которые в настоящее время включают расширенные возможности поиска и индексирования структурированных данных и метаданных, поэтому их часто называют хранилищами слабоструктурированных данных. В хранилище данных, ориентированном на документы, каждый документ инкапсулирует и кодирует метаданные, поля и любые другие варианты представления этой записи. Кроме того, что хранилища документов в какой-то степени аналогичны строке в реляционной таблице, одной из причин развития и популярности хранилищ документов является то, что большинство реализаций не применяют фиксированную (постоянную) схему. Передовой опыт утверждает, что группы документов должны быть логически связаны и содержать схожие данные, однако не требуется, чтобы они были одинаковыми или чтобы любые два документа содержали одни и те же поля. В этом заключается одна из причин, по которой хранилища документов достаточно популярны для хранения наборов данных с редко заполненными полями, поскольку в этом случае обычно возникает гораздо меньше накладных расходов, чем в традиционных реляционных системах управления базами данных, где в качестве записей фактически хранятся столбцы с нулевыми значениями. Группы документов в этих типах хранилищ обычно называются коллекциями, и, подобно хранилищам типа «ключ — значение», некоторые уникальные ключи имеют ссылки на каждый документ.

9.2.4.8 Функциональный компонент графовой базы данных

В то время как сайты социальных сетей, безусловно, стимулировали эволюцию графовых баз данных и способствовали повышению их наглядности (а также повышению эффективности обработки, о чем сказано далее), хранилища графов в течение нескольких лет были существенной составной частью многих предметных областей, начиная с военной разведки и борьбы с терроризмом и заканчивая планированием/навигацией маршрутов, а также семантическими сетями. Данные в графовых базах данных представлены в виде совокупности узлов, ребер и их свойств (характеристик). Для устранения неоднозначности сущностей и сравнения графов аналитика по графовым базам данных содержит результаты расчета основного (базового) кратчайшего пути и ранжирования страниц.

Фактически подходы к решению задачи построения графовых баз данных можно рассматривать как специализированную реализацию схемы хранения данных с двумя типами данных [узлы и отношения (их взаимосвязи)]. Кроме того, одним из наиболее значимых элементов анализа данных графа является определение местоположения узла или ребра графа, с которого должен начаться анализ. Для этого в большинстве графовых баз данных реализованы индексы свойств узлов или ребер. В отличие от реляционных и других подходов к хранению данных, большинство графовых баз данных, как правило, используют искусственные идентификаторы /псевдоключи или руководство для уникальной идентификации узлов и ребер. Это позволяет легко изменять атрибуты/свойства вследствие как фактических изменений в данных (кто-то изменил свое имя), так и обнаружения дополнительной информации (например, лучшего местоположения для некоторого элемента или события) без необходимости вносить изменения в указатели отношений в прямом направлении/обратном направлении.

Как правило, в распределенных архитектурах для обработки графов фрагменты графа сопоставляют системным узлам, после чего в системных узлах для передачи данных об изменениях в графе или полученных значениях используется подход на основе расчета путей в графе. Даже небольшие графы быстро превращаются в область больших данных, когда кто-то ищет шаблоны или расстояния на более чем одной или двух степенях разделения между узлами графа.

В зависимости от плотности графа такой подход может быстро вызвать комбинаторный взрыв применительно к количеству условий/шаблонов, которые будет необходимо проверить. Существует специализированный вариант реализации хранилища графов, известный как структура описания ресурсов (RDF), которая является частью семейства спецификаций консорциума World Wide Web (W3C). Эти спецификации часто напрямую ассоциированы с семантическим вебом и связанными с ним концептами. Тройки RDF, как известно, состоят из подлежащего (мистер *X*), предиката (живет в) и объекта (переулок пересмешника). Таким образом, набор троек RDF представляет собой ориентированный помеченный граф. Содержимое RDF-хранилищ часто описывают с использованием формальных языков описания онтологии типа OWL или языка RDF-схемы (RDFS), которые устанавливают семантические значения и модели базовых данных. Для поддержки лучшей горизонтальной интеграции [16] гетерогенных наборов данных предложены расширения концепции RDF, такие как структура описания данных (DDF) [17], которые добавили дополнительные типы данных для более эффективной поддержки семантической совместимости и анализа. В хранилищах графических данных в настоящее время отсутствуют какие-либо стандартизованные API-интерфейсы или языки запросов. Тем не менее консорциум W3C разработал язык запросов к данным, представленным по модели RDF SPARQL (SPARQL Protocol and RDF Query Language для RDF), который в настоящее время находится в статусе рекомендации, и есть несколько систем, таких как Sesame, которые становятся востребованными для работы с RDF и другими хранилищами данных, ориентированными на графы.

9.2.5 Функциональные компоненты ресурсного уровня

9.2.5.1 Общие положения

Функциональные компоненты ресурсного уровня включают в себя:

- компонент абстракции ресурсов и управления ими;
- компонент физических ресурсов.

9.2.5.2 Функциональный компонент абстракции и управления ресурсами

Функциональный компонент абстракции ресурсов и управления используется сервис-провайдером приложения больших данных для предоставления доступа к физическим вычислительным ресурсам посредством программной абстракции. Ресурсная абстракция необходима для обеспечения эффективного, безопасного и надежного использования базовой инфраструктуры. Функции ресурсной абстракции поддерживаются возможностями данного функционального компонента.

Примечание 1 — Когда система больших данных развернута в среде облачных вычислений, функции виртуализации ресурсов предоставляются средой облачных вычислений, как определено в [6].

Функциональный компонент абстракции ресурсов и управления позволяет сервис-провайдеру BDAP обеспечивать такие характеристики, как эластичность, объединение ресурсов и самообслуживание по требованию. Функциональный компонент виртуализации ресурсов и управления может включать программные элементы, такие как гипервизоры, виртуальные машины, виртуальное хранилище данных и обработку в режиме разделения времени.

Применительно к сети абстракция и управление ресурсами представляют собой средства, обеспечивающие передачу данных от одного компонента к другому на уровне инфраструктуры. Кроме того, сетевая инфраструктура может также включать автоматизированное развертывание, возможности предоставления или агентов, в том числе агентов мониторинга всей инфраструктуры, которые используются элементами управления/коммуникации, обеспечивающими реализацию конкретной модели.

Логическое распределение кластера/вычислительной инфраструктуры для выполнения вычислений может изменяться от грид-сети физических компьютеров в стойке до набора виртуальных машин, работающих у поставщика облачных услуг, или до сети слабосвязанных компьютеров, распределенных по всему миру и обеспечивающих доступ к неиспользуемым вычислительным ресурсам.

Примечание 2 — Гипервизор представляет собой часть компьютерного программного обеспечения, прошивки или оборудования, которое создает и запускает виртуальные машины. В таком виде гипервизор изначально работает на «голом железе» и управляет несколькими виртуальными машинами, состоящими из операционных систем (ОС) и приложений.

9.2.5.3 Функциональный компонент физических ресурсов

Функциональный компонент физических ресурсов представляет собой элементы, необходимые поставщикам приложений больших данных для запуска и управления предлагаемыми ими системами больших данных.

Физические ресурсы включают аппаратные ресурсы, такие как компьютеры (CPU и память), сети (маршрутизаторы, межсетевые экраны, коммутаторы, сетевые соединения и сетевые разъемы), компоненты хранения (жесткие диски) и другие элементы физической вычислительной инфраструктуры. Данные ресурсы могут включать те, которые находятся внутри облачных центров обработки данных (например, вычислительные серверы, серверы хранения и сети внутри дата-центров), и те, которые находятся за пределами центров обработки данных, — как правило, сетевые ресурсы, такие как сети центров обработки данных и опорные транспортные сети.

Для сети характеристики объема и скорости больших данных часто являются основными факторами в реализации внутренней и внешней связности сетевой инфраструктуры.

Физические ресурсы для вычислений — это физические серверы, которые выполняют и поддерживают программное обеспечение других компонентов системы больших данных. Вычислительная инфраструктура также часто включает базовые операционные системы и взаимодействующие сервисы, используемые для соединения кластерных ресурсов через сетевые элементы. Физические ресурсы для хранилища — это ресурсы, которые обеспечивают постоянство данных в системе больших данных. Инфраструктура хранения может включать любой ресурс от изолированных локальных дисков до сетей хранения данных (SAN) или сетевого хранилища.

Физические ресурсы — это физические ресурсы предприятия (электропитание, охлаждение), которые следует учитывать при создании экземпляра конкретного варианта системы больших данных.

Хотя компоненты ресурсов могут быть развернуты непосредственно на физических ресурсах или на виртуальных ресурсах, на определенном уровне все ресурсы имеют физическое представление. Физические ресурсы часто используются для развертывания нескольких компонентов, которые дублируются на большом количестве физических узлов, чтобы обеспечить так называемую горизонтальную масштабируемость. Виртуализация часто применяется для достижения эластичности и гибкости при распределении физических ресурсов и часто именуется «инфраструктура как услуга» (Infrastructure as a Service, IaaS) в рамках модели облачных вычислений.

Ресурсы, повышающие эффективность вычислений, хранения или скорости передачи систем больших данных, получили наименование ускорителей. Объем, разнообразие и скорость больших данных требуют более высокой и гибкой скорости обработки, чем при традиционном подходе.

Примечание — Например, ускорители для вычислений включают в себя следующее, но не ограничиваются этим: графический процессор, настраиваемую вентильную матрицу для ускорения вычислений путем ее программирования пользователем.

9.2.6 Многоуровневые функциональные компоненты

9.2.6.1 Общие положения

Многоуровневые функции включают ряд функциональных компонентов, которые предоставляют сервисы функциональным компонентам других уровней.

9.2.6.2 Функциональные компоненты уровня интеграции

9.2.6.2.1 Общие положения

Функциональные компоненты уровня интеграции предоставляют услуги для подключения функциональных возможностей компонентов на одном уровне или на нескольких уровнях.

Функциональные компоненты уровня интеграции могут включать, кроме всего прочего:

- среду обмена сообщениями (см. 9.2.6.2.2);
- среду управления состоянием (см. 9.2.6.2.3).

9.2.6.2.2 Функциональный компонент среды обмена сообщениями

Функциональный компонент среды обмена сообщениями предоставляет сервисы (например, в виде API) для маршрутизации и обмена сообщениями, включая, помимо прочего, надежную организацию очередей, передачу и получение данных между узлами в горизонтально масштабируемом кластере или компонентами в различные вертикальные уровни или между ними (см. рисунок 12). Например, сетевой ресурс на уровне ресурсов может отправлять информацию о своем состоянии на компоненты управления системой через предлагаемые API с помощью среды обмена сообщениями.

9.2.6.2.3 Функциональный компонент среды управления состояниями

Функциональный компонент среды управления состояниями используется другими функциональными компонентами для сохранения или поддержания состояния на узлах в распределенной среде, чтобы обеспечить согласованность состояний и постоянство во избежание сбоев ресурсов или системы. Информация о постоянном состоянии может быть введена в компоненты управления системой для мониторинга или управления ресурсами.

9.2.6.3 Функциональные компоненты уровня безопасности больших данных и конфиденциальности персональных данных

9.2.6.3.1 Общие положения

Компоненты безопасности больших данных и конфиденциальности персональных данных используются для облегчения взаимодействия в рамках эталонной архитектуры больших данных без ущерба для конфиденциальности персональных данных, конфиденциальности или целостности больших данных. Компоненты безопасности больших данных и конфиденциальности персональных данных тесно связаны со всеми функциональными компонентами через API.

Примечание — Компоненты безопасности больших данных и конфиденциальности персональных данных составляют фундаментальный аспект эталонной архитектуры. Они представляют собой геометрически охватывающие или сквозные основные компоненты, указывающие на то, что на все компоненты влияют соображения безопасности персональных данных. Таким образом, роль безопасности больших данных и конфиденциальности персональных данных корректно отображается по отношению к другим компонентам, но не расширяется до более мелких деталей, которые дают более точное представление, но скорее относятся к более детализированной эталонной архитектуре и безопасности больших данных и конфиденциальности персональных данных. Общие категории компонентов, реализованных для поддержки аспектов безопасности больших данных и для конфиденциальности персональных данных, приведены ниже.

Компоненты безопасности больших данных и конфиденциальности персональных данных взаимодействуют с некоторыми компонентами управления системой и обеспечивают их использование для сбора и отслеживания данных.

9.2.6.3.2 Функциональный компонент среды аудита

Функциональный компонент среды аудита используют другие компоненты для записи событий в системе. События могут включать пользователей, компоненты, задания и их действия, такие как запуск, остановка, доступ к данным, обновление данных и т. д. Для записи и сохранения данных эти компоненты достаточно часто применяют компоненты уровня платформы, но в целях безопасности могут сохранять данные и за пределами архитектуры больших данных. Журналы аудита, поддерживаемые этими компонентами, применяют для решения задач: отслеживания происхождения данных; восстановления данных/состояния в случае сбоя системного компонента, а также криминалистического анализа системного сбоя, вторжения.

9.2.6.3.3 Функциональный компонент среды аутентификации

Функциональный компонент среды аутентификации обеспечивает управление доступом к базовым данным и службам в других компонентах, а также доступ к системе в целом из внешних элементов. Аутентификация включает представление идентификатора (например, имени пользователя) и ключа или ключей доступа (например, пароля или сертификата), которые сверяются с хранилищем эталонных данных. Как правило, аутентифицируемый компонент, предоставляя идентификатор и ключ, связывается с компонентом, к которому он хочет получить доступ. Затем компонент, к которому осуществляется доступ, вызывает службы аутентификации и получает ответ о том, разрешить или отклонить доступ. Хотя в идеале сервисы аутентификации должны быть централизованы в одном компоненте, наличие нескольких компонентов на всех уровнях может включать разные уровни или компоненты, требующие разных компонентов аутентификации.

9.2.6.3.4 Функциональный компонент среды авторизации

Функциональный компонент среды авторизации обеспечивает поддержку функции сопоставления идентификатора пользователя или компонента с привилегиями, которые они имеют при доступе к ресурсам (как к данным, так и к обработке) в кластере.

Примечание — Примерами привилегий, которые могут быть применены к любому заданному ресурсу или элементу в кластере, являются чтение или доступ, запись, удаление, выполнение, обход и завершение.

Привилегии можно применять к ресурсу с разной степенью детализации. Например, многие платформы больших данных в настоящее время реализуют управление доступом на уровне полей/элементов, а не на уровне записей или файлов/наборов данных.

9.2.6.3.5 Функциональный компонент среды анонимизации

Функциональный компонент среды анонимизации поддерживает сохранение конфиденциальности или безопасности персональных данных путем запутывания одного или нескольких элементов данных, чтобы их невозможно было связать с другими элементами данных.

Примечание — Основным примером этого является анонимизация личной идентифицируемой информации (PII) о физических лицах для защиты их персональных данных. Подобные компоненты часто реализуют односторонние хеш-функции для создания уникальных значений, которые невозможно преобразовать к исходным значениям.

Службы авторизации используют для определения того, может ли пользователь или сервис получить доступ к исходным или собственным данным или доступ может быть только к закрытым данным.

9.2.6.4 Функциональные компоненты уровня управления системой

9.2.6.4.1 Общие положения

Функциональные компоненты уровня управления системой обеспечивают предоставление широкого спектра сервисов по установке, развертыванию, конфигурированию, мониторингу и настройке для функциональных компонентов на вертикальных уровнях, включая, помимо прочего:

- развертывание и конфигурацию (см. 9.2.6.4.2);
- мониторинг и оповещение (см. 9.2.6.4.3);
- мультитенантное управление ресурсами (см. 9.2.6.4.4);
- управление обеспечением высокой доступности (см. 9.2.6.4.5);
- функциональный компонент управления жизненным циклом больших данных (см. 9.2.6.4.6).

9.2.6.4.2 Функциональный компонент развертывания и конфигурации

Компоненты развертывания и конфигурации предоставляют функции для установки, развертывания и (повторной) конфигурации пакетов и сервисов на разных уровнях.

9.2.6.4.3 Функциональный компонент мониторинга и оповещения

Компоненты мониторинга и оповещения предоставляют функции для мониторинга состояния и производительности ресурсов и сервисов, развернутых на разных уровнях, а также отправки оповещений соответствующим компонентам управления при возникновении критических или опасных событий. Управление ресурсами по запросу, которое может использовать различные количества и различные типы ресурсов и услуг на любом уровне, необходимо для обеспечения масштабируемых вычислений и эластичности. Оповещения могут быть отправлены, например, компоненту управления ресурсами с несколькими арендаторами или компоненту управления высокой доступностью для запуска реконфигурации ресурса или сервиса.

9.2.6.4.4 Функциональный компонент управления мультитенантными ресурсами

Компонент управления ресурсами с несколькими арендаторами предоставляет функции для выделения изолированных ресурсов для разных арендаторов, запрашивающих услуги больших данных. Мультитенантность (многоарендность) представляет собой распространенный метод, популяризированный облачными вычислениями, позволяющий совместно использовать ресурсы и обеспечивать качество сервиса QoS среди разных арендаторов. Ресурсы, изолированные и предоставляемые арендаторам, могут охватывать: уровень ресурсов (например, центральный процессор и хранилища), уровень платформы (файловые системы или базы данных), уровень обработки (например, единая или гибридная платформа обработки) и уровень приложений для работы с большими данными (конкретные сервисы, предлагаемые арендаторам). В целях масштабируемости вычислений и обеспечения эластичности необходимы стандартные интерфейсы для управления ресурсами по запросу, которые могут использовать различные объемы и различные типы ресурсов и услуг на любом уровне.

9.2.6.4.5 Функциональный компонент управления обеспечением высокой доступности

Компоненты управления обеспечением высокой доступности реализуют функции для разработки политик, конфигурации и настройки статических или динамических сервисов, связанных с предоставлением избыточности, резервным копированием данных или ресурсов, запасным замещением и миграцией данных для восстановления после возникновения ошибок. В системе больших данных начиная от уровня ресурсов и до уровня промежуточного программного обеспечения могут возникать различные типы ошибок, например: сбои центрального процессора или хранилища, сбои отдельных узлов или кластера, программные ошибки, сбои питания или непредвиденное отключение устройств. Компоненты управления высокой доступностью могут получать входные данные от компонента мониторинга и оповещения и конфигурировать ресурсы или сервисы напрямую или через компонент управления ресурсами в режиме коллективной аренды.

9.2.6.4.6 Функциональный компонент управления жизненным циклом больших данных

9.2.6.4.6.1 Общие положения

Функциональные компоненты управления жизненным циклом больших данных предоставляют функции для управления жизненным циклом больших данных с момента поступления данных в систему через компонент импорта данных до их обработки или удаления из системы. Он может включать, помимо прочего, управление метаданными и управление качеством данных.

9.2.6.4.6.2 Функциональный компонент управления метаданными

Под управлением метаданными понимаются возможности оперирования метаданными, созданными на каждом этапе жизненного цикла больших данных — от приема, предварительной обработки, обработки, анализа, хранения до уничтожения или удаления.

Примечание — Управление метаданными необходимо для систем больших данных по следующим причинам:

- объем метаданных в настоящее время значительно больше, чем раньше, и постоянно растет;
- надлежащее управление метаданными играет существенную роль в процессе сбора и анализа данных, поскольку метаданные предоставляют информацию о том, как данные можно обрабатывать или использовать.

9.2.6.4.6.3 Функциональный компонент управления качеством данных

Управление качеством данных относится к установлению и развертыванию ролей, политик, действий и процедур, касающихся точности, целостности и полноты данных в течение жизненного цикла больших данных.

Примечание — Управление качеством данных имеет существенное значение для систем больших данных, поскольку низкое качество данных, например: когда данные являются неполными, ложными или устаревшими, может негативно повлиять на эффективность процессов интеллектуального анализа данных, на получение полезных результатов или привести к неправильным выводам.

Функции управления качеством данных обеспечивают взаимодействие с каждым вертикальным функциональным уровнем, поскольку на качество данных влияют все процессы импорта, интеграции, анализа, хранения, визуализации и потребления данных.

Приложение А
(справочное)

Сопоставление функциональных представлений при интеграции эталонной архитектуры больших данных с эталонной архитектурой других систем

Для конкретной системы больших данных пользовательское представление является единственным. Функциональное представление может быть применено к целевой системе или сервису. Например, на основе эталонной архитектуры облачных вычислений по ИСО 17789 определяется собственное функциональное представление для облачных вычислений. Если решение для больших данных реализовано в среде облачных вычислений, функциональное представление больших данных можно сопоставить с функциональным представлением облачных вычислений (см. рисунок А.1).

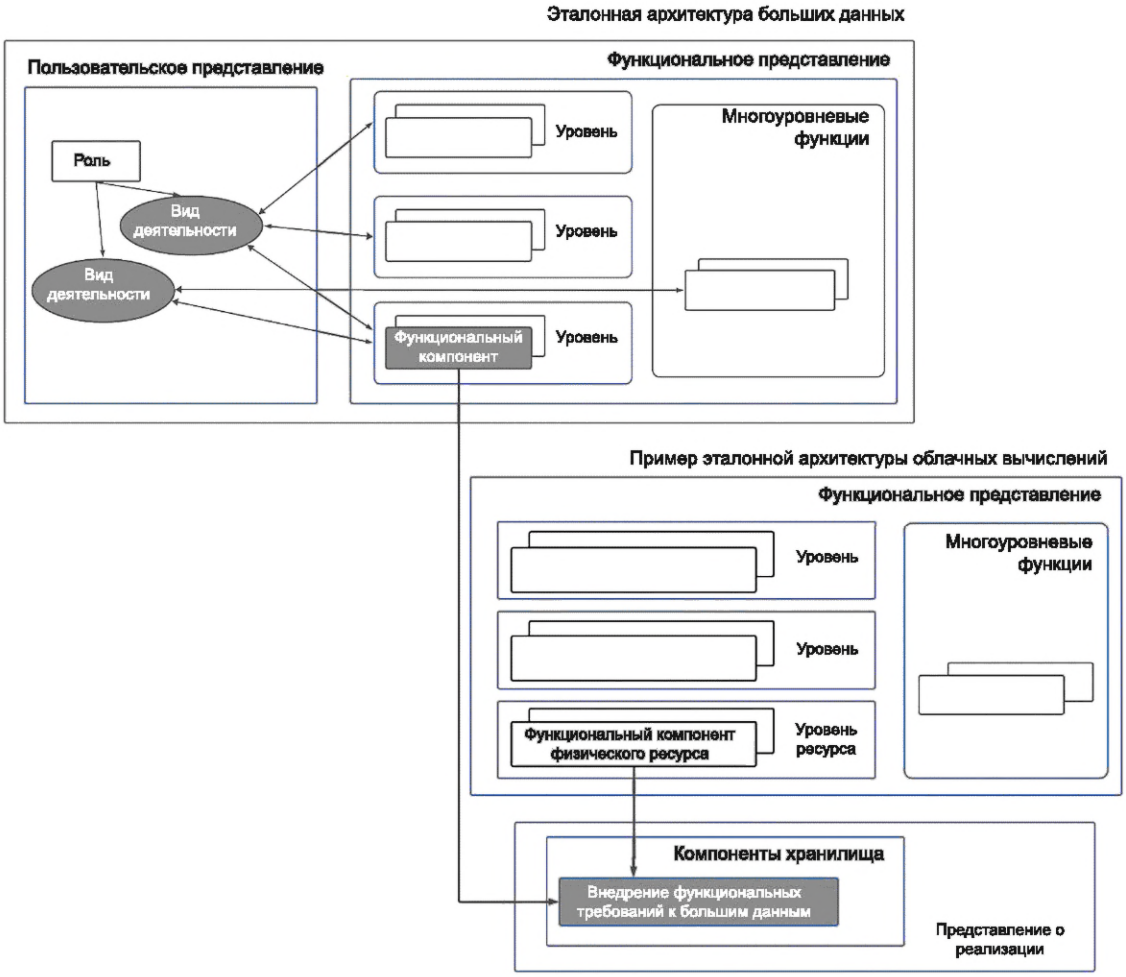


Рисунок А.1 — Сопоставление функциональных представлений при интеграции эталонной архитектуры больших данных с эталонной архитектурой облачных вычислений

Приложение В
(справочное)

Примеры взаимосвязей ролей в экосистеме больших данных

На рисунке В.1 представлено формальное описание эталонной архитектуры в терминах диаграммы классов UML. Необходимо отметить, что класс сервис-провайдера приложения больших данных имеет обратную ссылку «предоставления данных», чтобы учитывать возможность того, что сервис-провайдер приложения больших данных предоставляет данные другому сервис-провайдеру приложения больших данных, что, в свою очередь, позволяет предоставлять данные через цепь нескольких сервис-провайдеров приложения больших данных.

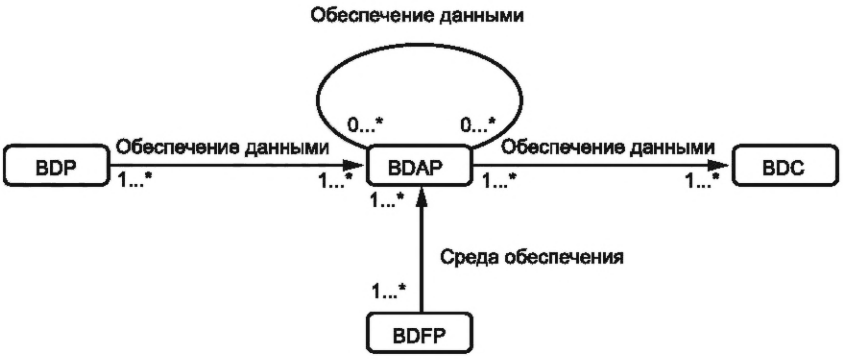


Рисунок В.1 — Диаграмма классов UML эталонной архитектуры

На рисунке В.2 представлены взаимосвязи между экземплярами ролей на основе вышеприведенной схемы UML. Данные каскадно проходят через нескольких сервис-провайдеров приложения больших данных.

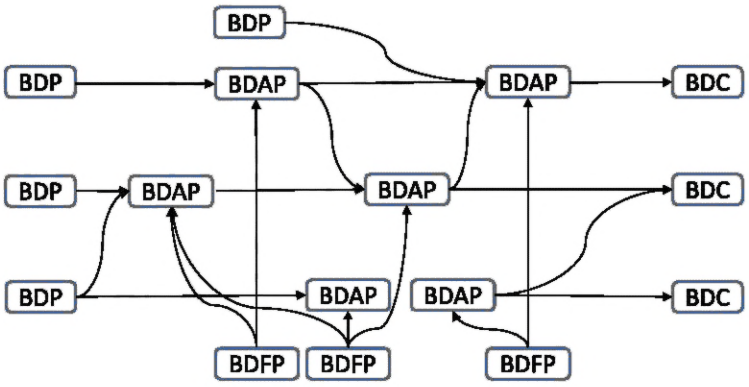


Рисунок В.2 — Пример взаимосвязей между экземплярами ролей больших данных

Приложение С
(справочное)

Основные понятия стратегического и оперативного управления данными, управления качеством данных в контексте больших данных

Основная цель данного приложения — дать характеристику основным понятиям: стратегического управления данными, управления качеством данных и оперативного управления данными в контексте больших данных. Указанные понятия, представленные в приложении, в основном перенесены из соответствующих документов Международной организации по стандартизации, упомянутых в приложении, а также из наиболее широко используемой и признанной литературы, полученной от конкретных профессиональных ассоциаций, таких как Ассоциация управления данными (DAMA, Data Management Association), Институт управления данными, Институт мастер-данных, Испанская ассоциация качества данных и информации (AECDI, Asociación Española para la Calidad de Datos y Información), Международная ассоциация, объединяющая профессионалов в области ИТ-аудита, ИТ-консалтинга, управления ИТ-рисками и информационной безопасности (ISACA, Information Systems Audit and Control Association) и Международная ассоциация качества информации и данных (IAIDQ, International Association for Information and Data Quality).

Представленный подход основан на идее одновременного рассмотрения «данных как актива организации» и «данных как продукта», поэтому данными следует управлять как активом и как продуктом. На рисунке С.1 показана взаимосвязь между тремя рассматриваемыми концепциями в рамках данного приложения.

Стратегическое управление данными представляет собой организационную функцию (т. е. совокупность организационных действий), которая гарантирует обеспечение того, чтобы данные, используемые в деловых процессах, создавали ценность и эффективно отвечали потребностям деятельности.

Оперативное управление данными представляет собой совокупность действий, направленных на поддержку жизненного цикла данных с технической точки зрения (сбор, описание, хранение, обработка и уничтожение) (DAMA, 2009).

Рисунок С.1 — Взаимосвязи между основными концепциями: оперативного управления данными, управления качеством данных и стратегического управления данными

В рамках стратегического управления данными решаются задачи планирования и определения стратегии организации, связанной с управлением данными, с целью обеспечения соответствия данных деловым потребностям. Для реализации этой стратегии необходимо оперативное управление данными за счет получения надлежащего набора ресурсов. Кроме того, в рамках оперативного управления данными решается задача реализации и развертывания стратегии управления данными с использованием набора индикаторов мониторинга процессов управления данными, определенных стратегией. Взаимосвязи между рассмотренными понятиями и функциями представлены на рисунке С.1.

Управление качеством данных представляет собой организационную функцию, направленную на подтверждение того, что данные имеют адекватный уровень качества в соответствии с потребностями деятельности. Адекватный уровень качества данных указывает на достоверность результатов деловых процессов, использующих данные.

В рамках стратегического управления данными определяют стратегию управления качеством данных, которая представляет собой совокупность ограничений и действий, направленных на достижение того, чтобы данные соответствовали требованиям к качеству согласно деловым потребностям. Кроме того, стратегическое управление данными обеспечивает предоставление вспомогательных ресурсов для управления качеством данных.

В рамках управления качеством данных определяют совокупность требований к качеству данных, показателей и критериев принятия решений, основанных на заданных ограничениях и действиях, чтобы эффективно контролировать и, при необходимости, улучшать качество данных.

Требования, а также способы измерения уровней качества данных передаются для оперативного управления данными, где они внедряются с применением метрик качества данных, разработанных в ходе управления качеством данных. Результирующие показатели качества данных используются в процессе управления качеством данных, который позволяет оценить, в какой степени удовлетворены деловые потребности организации в отношении качества данных.

Управление качеством данных предоставляет для стратегического управления данными набор показателей выполнения действий по обеспечению качества данных.

Стратегическое управление данными обеспечивает запрос у ИТ-подразделений, подразделений управления кадровыми ресурсами, а также подразделений управления корпоративными финансами необходимых ресурсов с целью гарантии выполнимости процессов управления качеством данных и оперативного управления данными.

Введенные понятия по возможности приведены в соответствие с опубликованными или разрабатываемыми стандартами ИСО в том случае, если такие документы существуют. С учетом вышеприведенного рассмотрим следующие нижеприведенные допущения:

- в связи с тем, что данные очень ценны для организаций, их можно рассматривать как актив и, следовательно, управлять ими соответствующим образом для достижения целей организации: в рамках этого утверждения можно считать, что:
  - данные о сделках как актив могут быть контекстуализированы в соответствии с принципами, представленными в ИСО 55000 [24],
  - данные о стратегическом управлении могут быть контекстуализированы в соответствии с принципами, представленными в ИСО/МЭК 58500 [21];
- в связи с тем, что данные можно прямо представить как исходный материал в виде необработанных данных и как результаты обработки данных, их можно рассматривать как продукт, а также обратить внимание на качество продукта. В рамках этой гипотезы можно считать, что:
  - управление качеством данных является восприимчивым к контекстуализации в соответствии с принципами, представленными в серии стандартов ИСО 8000,
  - определение характеристик качества данных как продукта, а также соответствующих показателей можно найти в ИСО/МЭК 25012 [9], ИСО/МЭК 25024 [10] и ИСО 8000-8 [30].

Понятия, извлеченные и адаптированные из перечисленного набора стандартов, могут быть дополнены другими существующими стандартами, касающимися конкретных вопросов управления данными ИСО 22745 [11], или стандартами, соответствующими проблематике управления данными или управления качеством данных в конкретных сферах деятельности ИСО 19157 [12], ИСО 13119 [13], ISO/TR 21707 [14], ISO/HL7 10781 [15].

Следует учесть, что использование терминов «стратегическое управление большими данными», «оперативное управление большими данными» и «управление качеством больших данных» не эквивалентно терминам: «стратегическое управление данными» в рамках проектов или экосистем больших данных, «оперативное управление данными» в рамках проектов или экосистем больших данных и «управление качеством данных» в рамках проектов или экосистем больших данных, поскольку понятия стратегического управления данными, управления качеством данных и оперативного управления данными выходят за рамки обычного использования данных.

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ISO 8000-2	IDT	ГОСТ Р ИСО 8000-2—2019 «Качество данных. Часть 2. Словарь»
ISO/TS 8000-60	—	*
ISO 8000-61	—	*
ISO/IEC 38500	IDT	ГОСТ Р ИСО/МЭК 38500—2017 «Информационные технологии. Стратегическое управление ИТ в организации»
ISO/IEC 38505-1	—	*
ISO/IEC TR 38505-2	—	*
ISO 55000	IDT	ГОСТ Р 55.0.01—2014/ИСО 55000:2014 «Управление активами. Национальная система стандартов. Общее представление, принципы и терминология»
ISO 55001	NEQ	ГОСТ Р 55.0.03—2021 «Управление активами. Системы менеджмента. Национальная система стандартов. Руководство по применению ISO 55001»
ISO 55002	—	*
ISO/IEC/IEEE 42010	IDT	ГОСТ Р 57100—2016/ISO/IEC/IEEE 42010:2011 «Системная и программная инженерия. Описание архитектуры»
ISO/IEC 20546	IDT	ГОСТ Р ИСО/МЭК 20546—2021 «Информационные технологии. Большие данные. Обзор и словарь»
ISO/IEC 17789	—	*

Примечание — В настоящей таблице использованы следующие условные обозначения степени соответствия стандартов:

- IDT — идентичные стандарты;
- NEQ — неэквивалентные стандарты.

* Соответствующий национальный стандарт отсутствует. До его принятия рекомендуется использовать перевод на русский язык данного международного стандарта.

Библиография

[1] Colella P., Defining software requirements for scientific computing. Slide of 2004 presentation included in David Patterson's 2005 talk. http://www.lanl.gov/orgs/hpc/salishan/salishan2005/davidpatterson.pdf

[2] Patterson D., Yelick K.., Dwarf Mind. A View From Berkeley. http://view.eecs.berkeley.edu/wiki/Dwarf_Mine

[3] United States Census Bureau, The "72-Year Rule." https://www.census.gov/history/www/genealogy/decennial_census_records/the_72_year_rule_1.html. Accessed March 3, 2015

[4] Apache Hadoop, Web HDFS REST API. https://hadoop.apache.org/docs/r1.0.4/webhdfs.html#FsURIvsHTTP_URL. Accessed Feb 24, 2017

[5] ISO/IEC 20546 Information technology — Big data — Overview and vocabulary

[6] ISO/IEC 17789 Information technology — Cloud computing — Reference architecture

[7] DoD Reference Architecture Description. https://dodcio.defense.gov/Portals/0/Documents/DIEA/RefArchiDescriptionFinalv118Jun10.pdf

[8] ISO/IEC 27002 Information technology — Security techniques — Code of practice for information security controls

[9] ISO/IEC 25012 Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model

[10] ISO/IEC 25024 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of data quality

[11] ISO 22745 Industrial automation systems and integration — Open technical dictionaries and their application to master data

[12] ISO 19157 Geographic information — Data quality

[13] ISO 13119 Health informatics — Clinical knowledge resources — Metadata

[14] ISO/TR 21707 Intelligent transport systems — Integrated transport information, management and control — Data quality in ITS systems

[15] ISO/HL7 10781 Health Informatics — HL7 Electronic Health Records-System Functional Model, Release 2 (EHR FM)

[16] Smith B., Malyuta T., Mandrick W.S., Fu C., Parent K., Patel M., (2012). Horizontal Integration of Warfighter Intelligence Data: A Shared Semantic Resource for the Intelligence Community. In Proceedings of the Conference on Semantic Technology in Intelligence, Defense and Security (STIDS), CEUR_. pp. 1—8

[17] Yoakum-Stover S., Malyuta T., Unified Integration Architecture for Intelligence Data.» Proceedings of DAMA International Europe Conference, London, UK. 2008

[18] ISO 8000-2 Data quality — Part 2: Vocabulary

[19] ISO/TS 8000-60 Data quality — Part 60: Data quality management: Overview

[20] ISO 8000-61 Data quality — Part 61: Data quality management: Process reference model

[21] ISO/IEC 38500 Information technology — Governance of IT for the organization

[22] ISO/IEC 38505-1 Information technology — Governance of IT — Governance of data — Part 1: Application of ISO/IEC 38500 to the governance of data

[23] ISO/IEC TR 38505-2 Information technology — Governance of IT — Governance of data — Part 2: Implications of ISO/IEC 38505-1 for data management

[24] ISO 55000 Asset management — Overview, principles and terminology

[25] ISO 55001 Asset management — Management systems — Requirements

[26] ISO 55002 Asset management — Management systems — Guidelines for the application of ISO 55001

[27] ISO/IEC/IEEE 42010 Systems and software engineering — Architecture description

[28] ISO/IEC 20547-4 Information technology — Big data reference architecture — Part 4: Security and Privacy

[29] ISO/IEC 27000 Information technology — Security techniques — Information security management systems — Overview and vocabulary

[30] ISO 8000-8:2015 Data quality — Part 8: Information and data quality: Concepts and measuring

УДК 004.8:006.354 ОКС 35.020

Ключевые слова: информационные технологии, эталонная архитектура, большие данные, пользовательское представление, функциональное представление, роль, подроль, сервис-провайдер, безопасность персональных данных, сквозные аспекты, управление данными, функциональная архитектура, функциональные компоненты, многоуровневая архитектура

Технический редактор *В.Н. Прусакова*
Корректор *М.И. Першина*
Компьютерная верстка *М.В. Малеевой*

Сдано в набор 31.10.2024. Подписано в печать 19.11.2024. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 4,65. Уч.-изд. л. 4,00.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
www.gostinfo.ru info@gostinfo.ru