

**О КОНЦЕПЦИИ ПОСТРОЕНИЯ И ВЫБОРА РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ
ИНФОРМАЦИОННО-ПОИСКОВЫХ СИСТЕМ****Введение**

Распределенные базы данных (БД) появились как следствие развития распределенных систем различного назначения и, в первую очередь, информационно-поисковых систем (ИПС), интегрирующих, как правило, множество баз данных, ориентированных на различные предметные области. Проектирование (и в том числе выбор) таких БД представляет собой весьма сложную задачу, поскольку, помимо проблем собственно интеграции и проектирования локальных классических БД, появились новые проблемы, характерные именно для распределенных БД. Это проблемы:

- оптимизации размещения распределенных БД по узлам сетевой структуры системы, что не поддерживается в настоящее время современными CASE – системами (Computer Aided System Engineering);
- сетевой масштабируемости БД для географически распределенных систем;
- проектирования и реализации (использования) коммуникационной среды системы памяти, как основы для погружения распределенной БД;
- оптимизации размещения данных в распределенных БД для обеспечения баланса загрузки;
- синхронизации доступа к данным и параллельной обработки запросов для обеспечения требуемой производительности системы;
- поддержания копий данных в нескольких узлах сети для сокращения пересылок при выполнении распределенных запросов;
- фрагментации объектов БД и расширения пространства поиска вариантов выполнения запросов;
- управления транзакциями (например, корректное завершение распределенных транзакций), многоуровневой защиты данных в БД и реализации возможности получения разнообразной по структуре и содержанию информации от различных типов источников.

В соответствии с этим сформировались новые требования к распределенным БД, появились новые технологии их проектирования. Возникла потребность в новой концепции создания и применения распределенных БД, которая бы учитывала все это разнообразие технологий и новых требований, с одной стороны, и непригодность для проектирования БД устаревших инструментальных средств, с другой. Концепции такого типа в настоящее время отсутствуют, несмотря на огромный объем публикаций по проблемам создания и применения распределенных БД.

В данной статье предлагаются основные положения новой концепции построения (выбора) распределенных баз данных, основными особенностями которой являются использование качественных и количественных признаков БД, определяемых различными технологиями их разработки; формирование так называемых типовых параметрических портретов БД; интеграция (в соответствии с многоуровневым представлением критериев качества) аппаратных и программных средств, соответствующих выделенным по специальной методике признакам, которые определяют значения параметров БД. При этом положения концепции детально не рассматриваются, поскольку каждое из них (например, установление соответствия между признаками БД и их поддерживающими аппаратно-программными средствами) представляет собой сложную самостоятельную проблему и может быть предметом отдельных исследований.

Для того, чтобы достаточно обоснованно подойти к изложению сущности концепции, определим место и особенности распределенных ИС и соответственно распределенных БД среди других разновидностей изделий своих классов.

Информационно – поисковые системы (ИПС) и проблемы их информационного обеспечения

С точки зрения структурной организации пользователей ИПС, можно выделить группу пользователей (или один пользователь); коллектив пользователей; территориально разнесенные группы пользователей.

Первая категория (одиночные ИПС) работает на автономном компьютере с разделением по времени одного рабочего места. Приложения создаются с помощью "настольных СУБД" (Clarion, Clipper, FoxPro, Paradox, dBase, MS Access) или с помощью файловой системы и диалоговой оболочки для ввода, редактирования и обработки данных.

Вторая категория (групповые ИПС) ориентирована на коллективное использование информации с применением локальной вычислительной сети ПК или многотерминальной централизованной системы. При разработке приложений используются многопользовательские "настольные СУБД", серверы БД для рабочих групп (Btrieve, NetWare SQL, Gupta SQLBase, Sybase Anywhere SQL, MS SQL Server, Progress, Informix-SE, Workgroup Oracle и др.) и соответствующие средства разработки или управления документами.

И, наконец, категория *распределенных пользователей или распределенных групп пользователей*, работающих на корпоративных ИПС, которые являются развитием групповых систем и могут поддерживать территориально разнесенные узлы или сети. Они, как правило, имеют иерархическую структуру и обеспечивают доступ к центральной или распределенной базе данных, а также к информационному фонду рабочей группы. Строятся на корпоративных SQL-серверах БД (Oracle7, Informix-OnLine, Informix-DSA, Sybase, CA-Ingress и др.) с применением соответствующих инструментальных средств.

На особенностях организации распределенных баз данных применительно к этой категории ИПС (как наиболее перспективной в настоящее время) остановимся более подробно.

Интеграция и размещение данных в ИПС. Различия в степени структурированности данных, доступных электронным способом, значительные. С одной стороны, данные, хранящиеся в традиционных реляционных и объектно-ориентированных БД, имеют вполне определенную структуру. С другой стороны, существуют как неструктурированные, так и полуструктурированные данные [1]. Аудио- и видеоизображения можно отнести к неструктурированным данным. Примером полуструктурированных данных может быть форматированный текст, HTML- страницы, данные в нетрадиционных форматах (например, в ASN.1), в формате XML и др.

При интеграции данных из разнородных источников необходимо привести их к общей структуре, что при большом количестве источников весьма трудоемкий процесс. При этом обобщенная структура будет очень сложной, а ее использование неэффективным. Поэтому наибольшую популярность приобрели два подхода к решению этой задачи: применение хранилищ данных и виртуальных хранилищ [2, 3].

Серьезная проблема размещения данных – преодоление перекосов в их распределении, что отрицательно влияет на баланс загрузки. В такой ситуации полезными могут оказаться гибридные архитектуры, узлы которых обладают разными вычислительными мощностями и объемами памяти. Другой подход состоит в дальнейшей декластеризации наиболее крупных разделов данных.

Еще один фактор, усложняющий задачу размещения данных, – это тиражирование для обеспечения высокой готовности. Для решения этой проблемы в последнее время было предложено несколько стратегий тиражирования.

Для того, чтобы сформировать новое знание на основе данных, применяют технологию интеллектуального анализа данных (ИАД или data mining), содержащую некоторое количество процедур, таких, как анализ проблемы и постановка задачи; подготовка данных; сбор данных; построение модели данных, её использование и сопровождение (наблюдение).

В настоящее время для анализа данных применяют нейронные сети, деревья решений, индукцию правил, а также дополнительные средства и методы: систему рассуждений на основе аналогичных случаев;

нечеткую логику; генетические алгоритмы; алгоритмы определения ассоциаций и последовательностей; логическую регрессию; эволюционное программирование и др. [4-7].

Развитие технологии ИАД связано, в первую очередь, с необходимостью аналитической обработки сверхбольших объемов информации, накапливаемой в современных хранилищах данных, необходимых для решения задач в области научных исследований, образовании, в работе правоохранительных органов, производстве, здравоохранении, в бизнес - приложениях и др.

Для решения указанных выше задач представления, структурирования, размещения и выдачи информации используются БД (с соответствующими СУБД), основные разновидности которых рассмотрены ниже.

Основные разновидности БД

Среди большого разнообразия БД можно выделить реляционные и постреляционные системы БД

На сегодняшний день наибольший интерес представляет вторая группа БД, поскольку они оперируют со сложноструктурированными объектами, что является характерным для таких прикладных систем, как САПР, системы искусственного интеллекта и т.д. Среди этой группы БД можно выделить [8]:

Активные базы данных. БД называется активной, если СУБД содержит продукционную систему, условия и действия которой не ограничиваются содержимым БД или прямыми действиями над ней со стороны пользователя.

Дедуктивные базы данных. БД состоит из двух частей: экстенциональной, содержащей факты, и интенциональной, содержащей правила для логического вывода новых фактов на основе экстенциональной части и запроса пользователя. Основным отличием дедуктивной СУБД от реляционной является то, что правила интенциональной части БД, а также запросы пользователей могут содержать рекурсию.

В том случае, когда набор правил дедуктивной БД становится большим и их невозможно разместить в оперативной памяти, возникает проблема управления хранением и доступом к ним во внешней памяти.

Для реализации дедуктивной СУБД обычно применяется реляционная система, которая выступает в роли хранителя фактов и исполнителя запросов, поступающих с уровня дедуктивной СУБД.

Темпоральные базы данных. Основные особенности темпоральных БД:

- для любого объекта данных, созданного в момент времени t_1 и уничтоженного в момент времени t_2 , в БД сохраняются (и доступны пользователям) все его состояния во временном интервале $[t_1, t_2]$;
- возможна выборка информации, хранившейся в базе данных в указанное время, в указанном временном интервале и т.д.;
- имеется возможность создавать версии отношений, и допускается их последующая модификация с учетом изменений основных вариантов.

Темпоральная БД (как и дедуктивная) – это надстройка над реляционной системой.

Мультибазы данных (интегрированные или федеративные системы). В мульти-БД применяются специальные способы именования для доступа к объектам локальных БД распределенных корпоративных систем. На глобальном уровне допускается только выборка данных, что позволяет сохранить автономность локальных неоднородных БД, распределенных в вычислительной сети. Как правило, для внешнего представления интегрированных и мульти-БД используется реляционная модель данных, хотя в последнее время все чаще предлагается применять объектно-ориентированные модели.

Распределенная база данных (DDB – Distributed Database) – это совокупность множества взаимосвязанных баз данных, распределенных в компьютерной сети. Система состоит из множества узлов приема запросов (query site) и непустого множества узлов данных (data site). Узлы данных обладают

средствами для хранения данных, а узлы приема запросов – нет; на них лишь выполняются программы, реализующие пользовательский интерфейс для доступа к данным, хранящимся в узлах данных. Узлы представляют собой независимые компьютеры, связанные сетью, а не процессоры, составляющие многопроцессорную конфигурацию. Важнейший отличительный признак – слабосвязанный характер среды, где каждый узел имеет собственную операционную систему и функционирует независимо [8, 9].

База данных физически распределяется по узлам данных при помощи фрагментации и репликации или тиражирования данных.

Распределенная база данных – это именно база данных, а не "коллекция" файлов, индивидуально хранимых на разных узлах сети. Распределенные данные представляют собой DDB, только если они связаны в соответствии с некоторым структурным формализмом (таким, как реляционная модель), а для доступа к ним имеется единый высокоуровневый интерфейс.

Распределенная БД не сводится по своим возможностям ни к распределенным файловым системам, ни к системам обработки транзакций. Обработка транзакций – только одна из функций, предоставляемых подобными системами.

Технология распределенных баз данных основывается на основополагающей для управления базами данных концепции: независимости данных на среду, где данные распределены и тиражированы по множеству компьютеров, связанных сетью.

Распределенные БД с интеллектуальными системами поиска информации. Одним из современных направлений искусственного интеллекта является создание интеллектуальной системы поиска релевантной информации с применением принципов построения мультиагентных систем (МАС), основной составляющей которых являются агенты [10]. Общая задача разбивается на подзадачи, а её решение осуществляется как композиция поведения агентов, реализующих выбор и выполнение последовательности доступных им действий, направленных на достижение собственных целей. При этом предполагается наличие механизмов адаптации агентов, алгоритмов их обучения, особенно, если речь идет о поиске релевантной информации в информационных средах. Особый интерес представляют генетические алгоритмы и эволюционные стратегии.

Функция обучения может быть реализована как непосредственно самим агентом за счет встроенного в них алгоритма, так и с применением общего алгоритма работы поисковой системы, что является универсальным и не зависит от конкретной реализации агента. В этом случае задача обучения агентов сводится к задаче оптимизации пространства поиска в наиболее перспективных областях, "богатых" релевантной информацией.

Объектно-ориентированные БД. Направление объектно-ориентированных баз данных (ООБД) определяется, прежде всего, потребностями практики: необходимостью разработки сложных информационных прикладных систем, для которых должно поддерживаться эффективное управление сложными структурами данных во внешней памяти.

Если за основу классификации принять особенности моделей данных, которые имеют те или иные БД, то можно выделить две основные группы [11]: чисто объектные ООБД и ООБД, основанные на модели сохраняемых объектов.

Чисто объектные в наиболее полной мере отражают все характерные черты объектной модели данных. Все создаваемые в них классы объектов по умолчанию сохраняются в базе данных. Как правило, такие системы поддерживают механизмы распределенных баз данных (transparent distributed database capabilities), многопользовательского доступа к БД, имеют встроенные средства разработки и администрирования.

ООБД, основанные на модели сохраняемых объектов, позволяют сохранять небольшие объекты (менее 20 байт), при этом в них сохраняются не сами объекты, а их идентификаторы. Такие системы обычно

имеют существенные ограничения на создание многопользовательских приложений и распределенных систем.

Языковая среда ООБД – это объектно-ориентированная система программирования, включающая средства работы с долговременными объектами. Среди языков и систем программирования большое влияние на ООБД оказал язык Smalltalk, в который была введена новая терминология, являющаяся наиболее распространенной в объектно-ориентированном программировании.

Создание и применение ООБД определяет при их проектировании ряд новых проблем в дополнение к проблемам создания других типов распределенных БД. Это проблемы:

- сохранения объектов БД и их многократного использования;
- обеспечения возможности сложной многопользовательской работы с произвольным типом данных, в том числе с неструктурированными документами (текст, мультимедиа) и сложно связанными данными;
- обеспечения возможности создания произвольных типов данных;
- обеспечения возможности эффективного взаимодействия с Internet;
- применения эффективных стратегий безопасности и защиты данных и использования сетей памяти для организации ООБД, интегрирующих аппаратные и программные платформы и др.

Естественно, что все эти проблемы определяются особенностями ООБД, определяющих их как класс распределенных БД, имеющих определенные диапазоны значений параметров, вполне конкретный набор функций и соответствующую область приложений (применений).

Основные параметры распределенных БД

Решение рассмотренных выше проблем для различных типов распределенных БД существенно сказывается на количественных и качественных показателях основных параметров, которые с достаточной степенью полноты характеризуют тот или иной класс (тип) БД. При этом, когда речь идет об оценке параметров БД конкретного типа (класса), то имеется в виду БД как цельное изделие, выполняющее конкретный набор функций. Для этого оно должно содержать не только собственно БД (запоминающее устройство со всеми атрибутами аппаратной и программной поддержки и его внутренним наполнением данными), но и систему управления БД, т.е. в классическом понимании – СУБД, поскольку, как показал анализ современных и перспективных БД, параметры БД и СУБД взаимосвязаны: конкретные значения параметров БД могут быть получены только при наличии конкретного типа СУБД и наоборот. С этой точки зрения оценку параметров и решение проблемы проектирования целесообразно осуществлять для БД, принимая во внимание наличие при этом соответствующей СУБД.

В качестве оценочных параметров БД обычно используются следующие: *производительность (эффективность) -- P, масштабируемость (расширяемость) -- M, надежность -- V, защита данных -- W, доступность -- Q, простота разработки приложений -- Z, уровень взаимодействия с пользователем -- X*. Совокупность этих параметров определим как характеристический набор Ψ изделия (системы):

$$\Psi = (P, M, V, W, Q, Z, X). \quad (1)$$

Производительность (P), как правило, обеспечивается за счет сочетания нескольких взаимно дополняющих решений, присутствующих в системе в целом, таких, как применение операционных систем, ориентированных на поддержку баз данных, параллелизм, оптимизация, баланс загрузки и др. Одним из способов повышения производительности серверов баз данных является распараллеливание: параллельные операции ввода-вывода с диском, параллельные утилиты и параллельная обработка запросов. Параллельные операции ввода-вывода с диском позволяют серверу эффективно использовать многомерные таблицы и разбиение таблиц. Параллельные операции-утилиты (сортировка, построение индексов, загрузка,

резервирование, восстановление) используют как параллельную обработку, так и параллельные обмены с диском [12].

Масштабируемость (M) – это свойство системы, которое допускает более высокую производительность, пропускную способность и т. д. путем добавления вычислительных ресурсов без изменения приложений и административного сопровождения. Существуют два способа расширения баз данных: горизонтальный и вертикальный. Горизонтальное расширение может быть получено, когда несколько серверов БД действуют независимо и делят рабочее пространство.

Вертикальное расширение достигается путем добавления к платформе дополнительных ресурсов, например, более быстрых процессорных элементов или дополнительных процессоров с целью уменьшения времени отклика или увеличения пропускной способности. Поддержка сервера баз данных для вертикального расширения не должна требовать включения дополнительных программных модулей, поскольку это увеличивает административные издержки.

В идеале параллельная и, в меньшей степени, распределенная БД обладает свойством линейной расширяемости и линейного ускорения. Под линейной расширяемостью понимается сохранение того же уровня производительности при увеличении размера базы данных и одновременном пропорциональном увеличении процессорной мощности и объема памяти [9]. Линейное ускорение означает, что с наращиванием процессорной мощности и объема памяти при сохранении прежнего размера базы данных пропорционально возрастает производительность. Причем при расширении системы потребуется минимальная реорганизация существующей базы данных.

Надежность (V). Это – защищенность БД от сбоев и способность восстанавливаться после них, обеспечивая доступ к данным как по чтению, так и по записи независимо от обстоятельств, включая выход из строя отдельных компонент. Такие системы определяются как отказоустойчивые. Они содержат, как правило, аппаратную избыточность и избыточность по данным.

В первом случае используются отдельные платформы, процессоры, сдвоенные диски и т. д. Обычным явлением является зеркалирование аппаратуры, при котором один диск используется как копия другого и защищает его от сбоев.

Избыточность по данным представлена в двух формах: программного зеркалирования и копирования, благодаря которым исключаются одиночные точки отказа. Отказ одного узла или сбоя на линии связи не приводит к выходу из строя всей системы.

Защита данных (W). Наиболее популярны два основных подхода к защите данных: применение к каждому защищаемому объекту БД набора допустимых привилегий; определение каждому пользователю некоторого набора прав доступа, например, основанного на использовании методов криптографии.

Доступность (Q). Основной характеристикой доступности БД является сопровождение их в оперативном режиме.

В идеальном случае утилиты сопровождения должны поддерживать непрерывные операции, с помощью которых система должна редуцировать или даже исключать плановые и неплановые неполадки. Физическое положение базы данных, реорганизация, управление памятью, архивирование протоколов и перезапуск системы должны иметь минимальное воздействие на приложения.

Простота разработки приложений (Z). Трудоемкость и простота разработки приложений определяются прежде всего наличием у разработчика развитых средств проектирования (специальных предпроцессоров и интерактивных средств разработки схем БД, возможности использовать средства многопользовательской разработки приложений, различных отладчиков и др.), дополнительных средств управления построением приложений, совершенного браузера БД, развитых средств настройки параметров, необходимого объема поставляемых библиотек объектов, а также полнотой технической документации, исходных текстов и других компонент, отмеченных в табл. 1– 5.

Уровень взаимодействия с пользователем (X) определяется наличием совокупности признаков (табл. 1– 5), обеспечивающих высокую эффективность общения пользователей с системой при реализации её основных функций и высоких показателей пользовательских характеристик системы в целом.

Новые требования и технологии проектирования распределенных БД

Проектирование комплексной по предметной направленности, интегрированной и, обычно, большой по размеру распределенной БД стало сложной задачей, поскольку при этом явно наблюдается интеграция различных структур данных (неформатированные элементы, ГИС-данные, мультимедийные, гипертекстовые распределенные данные и др.) и новые операции над данными (распределенная обработка, мультиплатформенность, многопротокольность, полнотекстовая обработка и др.).

Возникли новые требования к архитектуре корпоративной ИПС и, как следствие, – новые требования к распределенным БД, основные из которых следующие [13]:

- обеспечение доступа ко всем необходимым данным распределенных БД;
- применение наиболее типовых и перспективных архитектур БД и программных средств: хранилищ данных, оперативной аналитической обработки данных (OLAP), быстрой разработки приложений (RAD) для создания "ИС руководителя" (EIS), средств поддержки принятия решений (DSS) на основе хранилищ данных, OLAP и RAD/EIS и др.;
- применение методов логического вывода, нейронных сетей, нейрокомпьютеров и др.;
- применение единого интерфейса пользователя для работы с разными компонентами данных и приложений;
- постоянная актуализация понятийной модели для учета новых понятий, возникающих при изменении прикладных задач;
- динамическое администрирование распределенной корпоративной БД при изменении частоты использования, модификации структуры и изменении размещения данных и др.

Все это привело к модификации известных технологий проектирования БД (в частности, с применением CASE – систем), а также к поиску новых, сочетающих в себе перспективные направления и методы построения средств искусственного (машинного) интеллекта. К таким технологиям можно отнести:

1. Компонентную технологию проектирования предметно-ориентированных БД, основанную на формировании и использовании общей для интегрируемых компонент понятийной модели и допускающую работу пользователей через общие (в том числе для хранилища данных) интерфейсы [13].
2. Технологию проектирования (выбора), основанную на объектной парадигме, ориентированную на создание объектно-ориентированных БД (ООБД), для которых технологии предшествующих систем БД стали неудовлетворительными [11].
3. Технологию проектирования, основанную на сетевом принципе построения БД и сетевой модели организации данных [14].
4. Технологию проектирования БД, основанную на теории паттернов (шаблонов) [15].
5. Технологию проектирования, основанную на применении принципов построения многоагентных распределенных систем [10].
6. Другие перспективные технологии, в том числе технологии проектирования и применения компонентов БД информационных систем, а также реализации их отдельных функций (CORBA, COM, WEB – технологии и т.д.) [16 –18].

Основные положения концепции построения (выбора) ООБД

Предлагаемая концепция основывается на следующих положениях:

- 1) анализ и выделение доминирующих признаков БД, которые определяются использованием соответствующих технологий при их создании и применении;
- 2) учет влияния этих признаков (при выборе конкретной технологии) на качественные и количественные показатели распределенной БД;
- 3) интеграция (в соответствии с многоуровневой стратегией формирования критериев оценки параметров БД) средств поддержки для выбранной технологии;
- 4) использование в качестве инструментального средства для проектирования (выбора) БД системы информационно-аналитической поддержки разработчика.

При этом должны быть учтены следующие немаловажные факторы, которые оказывают существенное влияние на реализацию этой концепции, такие, как перечень проблем, подлежащих решению при создании (выборе) конкретного типа БД, в частности, ООБД; технические требования на изделие в целом; наличие необходимых инструментальных средств, а также существующие ограничения различного профиля на создание и применение БД (объем финансирования, таможенная политика, рынок аппаратных и программных средств и т.д.).

Рассмотрим несколько подробнее первое положение концепции, уделив особое внимание объектно-ориентированной технологии построения распределенных БД как весьма перспективной в настоящее время.

В результате анализа существующих информационных технологий и реализованных с их использованием распределенных БД были выделены (с учетом [11]) следующие группы доминирующих признаков, которые определяют основные параметры БД как законченного, готового к применению изделия:

1. Группа А. Характеристика модели структур данных и их представление в БД.
2. Группа В. Основные функции системы.
3. Группа С. Архитектурно-структурные компоненты БД для поддержки функций группы В и модели данных группы А.
4. Группа D. Особенности интерфейса пользователя (язык, категории запросов к БД, особенности разработки приложений и др.).
5. Группа F. Бизнес - факторы (производитель, поставщик, рынок).

Основные признаки каждой группы БД приведены в табл. 1 – 5 соответственно. При этом основной акцент сделан на класс объектно-ориентированных БД, т.е. на ООБД, которые в настоящее время занимают достойное место среди других разновидностей БД и для анализа которых имеется информация [11 –18].

Таблица 1. Набор основных признаков группы А

Описание признака	Влияние на характеристики БД
1. Наличие классов объектов	Увеличение гибкости проектирования, повышение производительности приложений.
2. Наличие идентификаторов объектов, возможности их повторного использования при удалении объекта	Повышение скорости доступа к сложно связанным данным
3. Наличие множества уникальных имен объектов	Уменьшение вероятности конфликтов имен объектов
4. Наличие набора полей представления данных и ссылок на другие объекты	Повышение скорости разработки приложений
5. Наличие независимости данных при их модификации различными клиентами	Разработка приложений: возможность параллельной работы группы программистов
6. Использование в качестве атрибутов текстовых и мультимедийных данных, а также ссылок на другие объекты	Увеличение производительности приложений со слабоструктурированными данными

7. Обеспечение возможности проверки типов данных на этапе трансляции программ	Ускорение модификации и простота отладки приложений
8. Наличие отношений, в том числе семантических отношений типа часть некоторого целого	Повышение эффективности выполнения операций над сложно связанными данными, в том числе – групповых операций над объектами
9. Наличие типов наследований, управление конфликтами при множественном наследовании	Сокращение времени разработки приложений
10. Применение модели агентов распределенной БД, построенной по принципам многоагентной системы (МАС)	Обеспечение повышения надежности, производительности разработки приложений и реализации процедур в соответствии с парадигмой МАС
11. Применение модели многокомпонентных объектов (Component Object Model –COM), а также распределенной COM (DCOM)	Обеспечение взаимодействия приложений посредством объектного интерфейса на языке описания, который не зависит от языка реализации объекта
12. Наличие неструктурированных документов (текст, мультимедиа) и данных с многомерными связями (например, кодов HTML-страниц)	Для хранения в БД требуется создание WEB-архитектур ООБД (WEB-интерфейса)
----- n	-----

Таблица 2. Набор основных признаков группы **B**

Описание признака	Влияние на характеристики БД
1. Реализация механизмов доступа к неограниченным данным	Возможность манипулирования большими объектами, которые не размещаются в адресном пространстве приложений
2. Наличие структурной и логической целостности данных	Возможность доступа из приложений к схеме данных, которая не реализуется традиционными способами
3. Реализация одновременного доступа к данным	Возможность одновременного доступа нескольких пользователей, а также размещения в БД частично модифицированных данных
4. Работоспособность БД при аппаратных и программных сбоях	Обеспечение параметров надежности ООБД
5. Применение нескольких схем и типов транзакций	Возможность управления конкуренциями (конflikтами) в распределенных приложениях
6. Реализация различных уровней блокировок	Повышение производительности в многопользовательском режиме
7. Резервное копирование в процессе использования БД	Увеличение производительности и надежности
8. Реализация выгрузки и загрузки данных в понятном пользователю формате файла	Повышение надежности хранения информации, упрощение её переноса в прикладные программы
9. Наличие пассивной или активной модели сообщений	Обеспечение приложениям возможности периодически запрашивать БД об изменении объектов или событий
10. Применение технологий	Оптимизация поиска данных в БД по различным критериям

индексирования, кэширования и b - деревьев	путем введения дополнительных данных - индексов
11. Реализация различных стратегий безопасности	Повышает безопасность БД от несанкционированного доступа к данным
12. Распределенность и параллельность вычислений и запросов к БД, возможность перестройки связей между агентами в MAC поиска информации	Возможность эффективной обработки больших объемов информации баз данных, электронных библиотек и каталогов, повышение надежности системы за счет перестройки связей и передачи функций одних агентов другим
13. Реализация стандартизованных технологий COM (DCOM), определяющих соглашения, сервисы описания, использования и коммуникации объектов	Возможность работы клиентов и серверов на распределенных платформах, в том числе Интернет и Интранет
14. Наличие функций поддержки мультимедиа и возможности работы с многосвязными данными	Простота доступа к документам, хранимым в БД, в сетях Интернет и Интранет
----- h	-----

Таблица 3. Набор основных признаков группы С

Описание признака	Влияние на характеристики БД
1. Применение сервера объектов распределенной клиент-серверной архитектуры	Высокая производительность: единица обмена – объект, кэширование и блокировки – на уровне объектов
2. Применение сервера страниц распределенной клиент-серверной архитектуры	Единица обмена – страница объектов, методы реализуются на клиенте, блокировки – на уровне страниц
3. Применение сервера файлов распределенной клиент-серверной архитектуры	Простой сервер БД, низкая эффективность системы
4. Поддержка технологии обмена данными между сервером и клиентом	Возможность перемещения связанных объектов, существенное повышение эффективности многопользовательских приложений
5. Поддержка технологии объединения объектов в кластеры	Сокращается время на поиск и пересылку близких объектов
6. Поддержка технологии масштабируемости (расширяемости)	Реализация нескольких вычислительных парадигм и переносимости СУБД на различные платформы
7. Поддержка эффективных механизмов интеграции распределенных данных	Возможность создания распределенных приложений, работающих на разных аппаратных и программных платформах
8. Поддержка активных и пассивных СУБД	Возможность сохранения методов объектов в БД, что значительно сокращает трафик компьютерной сети
9. Поддержка технологии композиции поведения агентов, решающих параллельно свои локальные задачи для	Оптимальное использование ресурсов системы, повышение производительности за счет разбиения общей задачи на подзадачи и распределения их по агентам

достижения общей цели	
10. Поддержка технологии интеграции концепций агентов и распределенных объектов на основе многоуровневой структуры взаимодействия агентов	Упрощается взаимодействие пользователя с распределенными компонентами сети при применении Интернет и Интранет решений для создания пользовательского интерфейса
11. Наличие для объектной БД средств взаимодействия с Интернет и Интранет	Обеспечивается высокопроизводительная работа с многосвязными данными
----- s	-----

Таблица 4. Набор основных признаков группы D

Описание признака	Влияние на характеристики БД
1. Наличие языков определения (Data Definition Language) и манипулирования (Data Manipulation Language) данными	Упрощается разработка приложений
2. Вычислительная полнота языков БД	Возможность языка БД своими собственными средствами решать все вычислительные задачи, что упрощает разработку приложений
3. Интеграция языка БД со стандартным языком программирования. Поддержка языков: C++, Smalltalk и др. или их расширений. SQL – совместимость	Совместимость со стандартным языком программирования, что упрощает разработку приложений и перенос существующего задела в среду объектной СУБД
4. Поддержка независимости данных: инкапсуляция и поддержка производных атрибутов	Способность изменять схему БД без нарушения доступа к данным со стороны приложения
5. Поддержка стандартов, например, близость интерфейса СУБД положениям стандарта ODMG-93	Возможность свободного перемещения приложений в среду других объектных СУБД
6. Поддержка ассоциативных запросов, объектных расширений SQL - запросов	Выборка разнородных объектов по различным критериям, произвольный тип результата запроса (значение, объект, коллекция объектов), генерация отчетов в виде текстов и т.д.
7. Поддержка специальных процедур, обеспечивающих независимость данных, и исключение их ошибочного искажения прикладными программами	Использование специальных методов поддержания независимости данных: применение поисковых процедур в сохраняемых классах, поддержка объектных расширений SQL - запросов и т.д.
8. Наличие средств просмотра и анализа вызванных ранее запросов	Повышает эффективность приложений, например, путем кэширования поисковых запросов и соответствующих им результатов поиска
9. Поддержка персональных БД, рабочих групп пользователей	Возможность перемещать объекты на рабочие места пользователей, возможность создания сложных многопользовательских систем
10. Доступ к схеме БД из приложений	Возможность модификации схемы БД на этапе выполнения прикладных программ, возможность создания динамически

	расширяемых приложений
11. Применение языка KQML (Knowledge Query and Manipulation Language)-языка общений между агентами, формата для обмена знаниями-KIF (Knowledge Interchange Format)	Унификация контекста всего процесса общения агентов для использования преимуществ каждого агента в составе MAC
12. Возможность адаптации и обучения агентов в распределенной БД MAC	Улучшение качества пользовательского интерфейса за счет повышения уровня машинного интеллекта
13. Применение HTTP-протокола взаимодействия браузера (WEB-клиента) с HTTP-сервером, общего шлюзового интерфейса-CGI и интерфейса приложений Интернет-сервера ISAPI	Обеспечение динамического формирования и наполнения данными HTTP-файлов, "интерактивность" WEB-сайта, возможность вести диалог с пользователем
----- k	-----

Таблица 5. Набор основных признаков группы F

Описание признака	Влияние на характеристики БД
1. Общая характеристика компании-поставщика распределенной БД (СУБД)	Качественное и долговременное сопровождение программного продукта
2. Маркетинговая политика компании. Стратегия продажи продуктов	Обеспечение высоких технических характеристик продуктов, возможность их приобретения для модификации системы
3. Технический уровень продукта, его место и популярность на рынке	Сокращение трудоемкости и затрат на освоение и применение продукта
4. Полнота технической документации, наличие примеров применения продукта	Сокращение времени на освоение продукта, повышение эффективности разработки приложений
5. Участие компании в разработке стандартов и следование тенденциям государственной политики по стандартизации	Важный фактор, влияющий на совместимость и масштабируемость системы в целом
6. Наличие лицензий и патентов на отдельные компоненты и продукт в целом	Определение защищенности продукта и влияния на его стоимость, в том числе услуги по сопровождению и обучению
----- -- t	-----

Таким образом, каждая группа (согласно табл. 1 – 5) может быть представлена соответствующими наборами доминирующих признаков:

$$\begin{aligned}
 A &= (a_1, a_2, \dots, a_n); \\
 B &= (b_1, b_2, \dots, b_h); \\
 C &= (c_1, c_2, \dots, c_s); \\
 D &= (d_1, d_2, \dots, d_k);
 \end{aligned}
 \tag{2}$$

$$F = (f_1, f_2, \dots, f_i).$$

По второму положению концепции считаем целесообразным отметить следующее.

Анализ информации, приведенной в указанных выше таблицах, показывает, что каждому параметру множества Ψ выражения (1) соответствует набор признаков G_j ($j = p, m, v, w, q, z, x$), элементы которого могут принадлежать различным группам параметров множества Y , где $Y \supset (A, B, C, D, F)$. Совокупность таких наборов G_j , согласно содержательной сущности входящих в него элементов (признаков) применительно к рассматриваемому случаю, можно представить в виде, показанном в табл. 6.

Таблица 6. Наборы доминирующих признаков G_j для параметров БД согласно (1)

Параметры БД	Набор доминирующих признаков G_j для данного параметра
$P = \varphi_1(G_p)$	$G_p = (a_1, a_4, a_5, a_6, a_8, a_9, a_{10}; b_6, b_7, b_{12}; c_1, c_4, c_9, c_{11}; d_8; f_4)$
$M = \varphi_2(G_m)$	$G_m = (b_{13}; c_2, c_6; d_4, d_5, d_{10}; f_2, f_5)$
$V = \varphi_3(G_v)$	$G_v = (a_3, a_9, a_{10}; b_4, b_5, b_7, b_8, b_{12}; c_9, d_7)$
$W = \varphi_4(G_w)$	$G_w = (b_{11}; f_6)$
$Q = \varphi_5(G_q)$	$G_q = (a_7, a_{11}; b_1, b_3, b_8, b_9, b_{10}, b_{12}, b_{13}, b_{14}; c_1, c_4, c_6, c_8; d_5, d_6, d_8, d_9, d_{10}, d_{11}, d_{12}, d_{13}; f_1, f_2, f_3, f_4, f_5, f_6)$
$Z = \varphi_6(G_z)$	$G_z = (a_1, a_5, a_7, a_8, a_{11}; b_1, b_3, b_5, b_6, b_8, b_9, b_{12}, b_{13}, b_{14}; c_4, c_5, c_7, c_8, c_9, c_{10}, c_{11}; d_1, d_2, d_3, d_5, d_6, d_7, d_8, d_{11}, d_{12}; f_3, f_4, f_5)$
$X = \varphi_7(G_x)$	$G_x = (a_2, a_{11}; b_1, b_2, b_3, b_8, b_{10}, b_{12}, b_{13}, b_{14}; c_4, c_7, c_{10}; d_1, d_2, d_3, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}, d_{13}; f_3, f_5)$

Отметим, что каждая компонента множества G_j вносит свой конкретный “вклад” в формирование значений соответствующих параметров множества Ψ . Причем анализ табл.6 показывает, что, если G_j для соответствующего параметра содержит элементы всех (или по крайней мере большинства) групп параметров A, B, C, D, F, то это значит, что данный параметр множества Ψ является доминирующим для БД, построенной по рассматриваемой технологии, так как его качественные и количественные показатели сформированы на всех уровнях организации и применения БД: на уровне представления данных, функциональном уровне, архитектурно-структурном, уровне пользовательского интерфейса и уровне бизнес - процесса. Причем среди выделенного таким образом множества параметров наибольший приоритет имеет тот, мощность множества G_j которого является наибольшей. В частности, для случая, отраженного в табл.6, доминирующими (в порядке уменьшения приоритетов) являются параметры: простота разработки приложений, уровень взаимодействия с пользователем, доступность, производительность, надежность, масштабируемость.

Определим совокупность параметров множества Ψ , размещенных вместе с соответствующими множествами $G_j^{(\gamma)}$ в порядке уменьшения их приоритета как *параметрический портрет* БД, реализованной с применением γ -ой технологии. Такого вида параметрический портрет будем называть *типовым*, поскольку он формируется специалистами высшей категории на основе уже известных признаков,

которые проявились при создании и применении БД за счет использования конкретной γ -ой технологии. Есть основания утверждать, что такие типовые параметрические портреты могут быть созданы для БД, построенных с использованием указанных технологий, в частности объектно-ориентированной технологии, многоагентной технологии и др.

Например, параметрический портрет ООБД, согласно табл. 1 – 6, будет иметь вид, отображенный в табл. 7.

Таблица 7. Параметрический портрет ООБД, построенный по данным табл. 1 – 6

Параметры ООБД	Набор доминирующих признаков G_j для данного параметра
$Z = \varphi_6(G_z)$	$G_z = (a_1, a_5, a_7, a_8; b_1, b_3, b_5, b_6, b_8, b_9; c_4, c_5, c_7, c_8; d_1, d_2, d_3, d_5, d_6, d_7, d_8; f_3, f_4, f_5)$
$Q = \varphi_5(G_q)$	$G_q = (a_7; b_1, b_3, b_8, b_9, b_{10}; c_1, c_4, c_6, c_8; d_5, d_6, d_8, d_9, d_{10}; f_1, f_2, f_3, f_4, f_5, f_6)$
$X = \varphi_7(G_x)$	$G_x = (a_2; b_1, b_2, b_3, b_8, b_{10}; c_4, c_7; d_1, d_2, d_3, d_5, d_6, d_7, d_8, d_9, d_{10}; f_3, f_5)$
$P = \varphi_1(G_p)$	$G_p = (a_1, a_4, a_5, a_6, a_8, a_9; b_6, b_7; c_1, c_4; d_8; f_4)$
$V = \varphi_3(G_v)$	$G_v = (a_3, a_9; b_4, b_5, b_7, b_8; d_7)$
$M = \varphi_2(G_m)$	$G_m = (c_2, c_6; d_4, d_5, d_{10}; f_2, f_5)$
$W = \varphi_4(G_w)$	$G_w = (b_{11}; f_6)$

Параметрический портрет БД, соответствующий γ -ой технологии, обозначим как

$$\Omega(\gamma) \supset (\theta_\gamma, G_\theta),$$

где θ_γ – множество (набор) параметров, по которым оценивается БД, реализованное с применением γ -ой технологии; G_θ – множество доминирующих признаков, определяющих значения параметров набора θ_γ .

Применительно к ООБД для рассматриваемого примера (табл. 1 – 7) имеем

$$\Omega(\text{ООБД}) \supset (\theta_{(\text{ООБД})}, G_\theta),$$

где $\theta_{(\text{ООБД})} = (Z, Q, X, P, V, M, W)$; $G_\theta = (G_z, G_q, G_x, G_p, G_v, G_m, G_w)$.

Обозначим множество параметров, требования к которым определены техническим заданием (ТЗ) на разработку (или выбор) БД, в виде $\theta_{(\text{ТЗ})}$. Принимая во внимание коэффициенты важности каждого из параметров этого набора, которые можно определить различным способом (например, экспертным путем), это множество параметров преобразуем в другое $\theta^{y_{(\text{ТЗ})}}$, элементы которого упорядочены (как и в типовых параметрических портретах БД) по убыванию их критериев важности. Тогда, сравнивая поочередно множество $\theta^{y_{(\text{ТЗ})}}$ с множествами θ_γ для всех значений γ на совпадение по приоритетности их нескольких (например, пяти наиболее важных) первых членов, можно сделать выбор предпочтительной технологии построения БД с точки зрения наиболее вероятного соответствия значений параметров вновь создаваемого (выбираемого) БД техническому заданию. Действительно, γ -ая технология является предпочтительной для

использования, если типы параметров и расстановка по приоритетности элементов множества θ_y в базовом параметрическом портрете БД, построенной по данной технологии, полностью совпадают с типами параметров и расстановкой по важности соответствующих элементов множества $\theta^y_{(T3)}$.

Третье положение концепции имеет прямое отношение к интеграции и оценке средств поддержки для реализации выбранной технологии построения БД, что, в принципе, позволяет уточнить ожидаемые значения параметров проектируемого БД и тем самым приемлемость предварительно выбранной технологии. Для этого из множества существующих аппаратных и программных средств, каждому из которых при составлении табл. 1 – 5 должно быть поставлено в соответствие конкретный признак множества Y , интегрируются при создании ООБД только те, которые обеспечивают реализацию и поддержку выделенных в табл. 7 соответствующих признаков. Например, в табл. 3 признакам c_1, c_2, c_3 группы С поставлено в соответствие клиент - серверная архитектура с применением сервера объектов, сервера страниц и сервера файлов (в табл. 1 – 5 соответствующие аппаратно-программные средства в явном виде для всех признаков не обозначены). Предполагается, что такие соответствия определяются высококвалифицированными специалистами в области создания и применения БД и заносятся в отдельный столбец при составлении таблиц признаков всех групп.

Сам же процесс интеграции определенных таким образом аппаратных и программных средств фактически сводится к выбору оптимального сочетания аппаратных и программных платформ (например, по методике, изложенной в [19]), в совокупности обеспечивающих реализацию требуемого набора функций и значений параметров распределенной БД, удовлетворяющих требованиям ТЗ. При этом должны быть учтены:

- тип, архитектурно-структурная организация, функции и параметры системы управления распределенной базой данных (СУБД), как если бы она была выполнена по той же технологии, что и проектируемая (выбираемая) БД, поскольку требуемые характеристик БД не могут быть получены без применения соответствующей СУБД и наоборот;
- тип, архитектурно-структурная организация и параметры коммуникационной сети памяти как физической среды, в которой происходят задержки управляющих сигналов (например, сигналов запроса к БД) и полученных в результате выполнения запросов данных.

Если значимость СУБД может быть учтена сразу же при составлении таблиц групп признаков БД (например, для ООБД предполагается применять соответствующую объектно-ориентированную распределенную СУБД, так как это принято при составлении табл. 1 – 5), то значимости коммуникационной сети должно быть уделено особое внимание, так как задержки прохождения сигналов в ней могут быть настолько существенными, что они могут свести на нет все положительные признаки проектируемой БД, “добытые” с большими усилиями при использовании новых технологий.

Процесс интеграции должен контролироваться на конкретных этапах путем сравнения полученных в результате интеграции значений параметров с соответствующими критериями оценки, которые в общем случае могут быть сформированы в виде дерева критериев, например, по методике, изложенной в [20]. При этом в основу построения дерева критериев положено так называемое дерево сравниваемых объектов, где в качестве объектов на соответствующих уровнях выступают варианты аппаратных и программных средств различной сложности, начиная от построенных по различным технологиям вариантов БД в целом (на самом верхнем уровне), серверов, коммуникационных сетей, разновидностей памяти (например, архивной и др.), необходимого для всех этих средств программного обеспечения и т.д. – для промежуточных уровней и кончая компонентами изделий (интерфейсными блоками, материнскими платами, чипсетами и др.) – на самом нижнем уровне. Степень детализации интегрируемых компонентов изделия и, следовательно, глубина дерева, как правило, определяется уровнем квалификации разработчика.

Соответственно дерево критериев содержит такое же количество уровней, при этом на самом нижнем уровне используются так называемые первичные критерии, по которым оцениваются характеристики элементов конечной детализации дерева объектов, на последующих уровнях – комплексные критерии и на самом верхнем уровне – один или несколько глобальных критериев.

При проектировании такого типа систем удобнее для оценки альтернативных вариантов применять ограниченное количество комплексных параметров, в качестве которых, например, могут выступать: время разработки (реализации) тестовых приложений; степень защищенности и надежности ООБД; уровень удобства настройки, эксплуатации и сопровождения системы и др. Возможно также применение глобального параметра (критерия) системы в целом, например, стоимость единицы производительности при разработке тестовых приложений и др.

Четвертое положение концепции – использование в качестве инструментального средства системы информационно-аналитической поддержки разработчика (СИАП) для проектирования (выбора) БД продиктовано необходимостью анализа и обработки огромного объема информации о предметных областях, высокими требованиями к квалификации разработчика БД, отсутствием каких-либо корректных зависимостей между параметрами изделия и степенью их влияния на архитектурно-структурные решения организации БД и их составляющих компонентов, что позволило бы “вручную” рассчитать требуемые параметры и определить оптимальное сочетание аппаратно-программных средств при построении БД.

Укрупненная схема реализации основных положений предложенной концепции построения (выбора) распределенной БД приведена на рис. 1.

Следует отметить, что установление зависимостей между различными параметрами является весьма сложной и до сих пор не разрешенной проблемой. Имеющиеся варианты такого рода зависимостей (например, быстродействия оперативной памяти от её ёмкости) носят субъективный характер, являются приближенными и распространены лишь на определенный тип изделия, поэтому имеют весьма ограниченную область приложений. Установленные же закономерности (например, закон Гроша, закон Мура и закон Амдала) основываются на различных допущениях и предположениях и пригодны лишь для весьма приближенных оценок тех или иных параметров СВТ. Так, закон Амдала, связывающий прирост производительности вычислителя от увеличения количества параллельно работающих процессоров и количества участков алгоритма, которые могут быть распараллелены, допускает, что процессоры должны быть одинаковыми и что пропускная способность коммуникационной среды, связывающей эти процессоры, – идеальная, т.е. не имеет задержек [21].

Существенно облегчить процедуру решения такой задачи возможно, если лицо, осуществляющее проектирование (выбор), будет иметь соответствующую информацию, заранее подготовленную квалифицированными специалистами в области проектирования и практического использования БД различных типов, которая в структурированном виде (например, в виде таблиц) представлена в БД инструментальной системы, например, системы информационно-аналитической поддержки процессов проектирования. В качестве такой информации должны выступать качественные и количественные показатели БД, их отличительные признаки по следующим направлениям:

1. Описание возможных разновидностей структурной организации пользователей (один или группа пользователей; коллектив пользователей; территориально разнесенные группы пользователей и др.).

2. Доминирующие структуры используемых данных, модели и методы их представления, размещения и интеграции.

3. Основные характеристики и особенности перспективных технологий проектирования БД (компонентная технология; технологии, основанные на объектной парадигме, на сетевом принципе построения БД и сетевой модели организации данных, теории паттернов (шаблонов), применении принципов построения многоагентных распределенных систем и др.).

4. Архитектурно-структурная организация БД, построенных по различным технологиям (распределенные БД, распределенные с интеллектуальными системами поиска информации, объектно-ориентированные БД и др.), их отличительные признаки и параметры. Перспективные направления их развития.

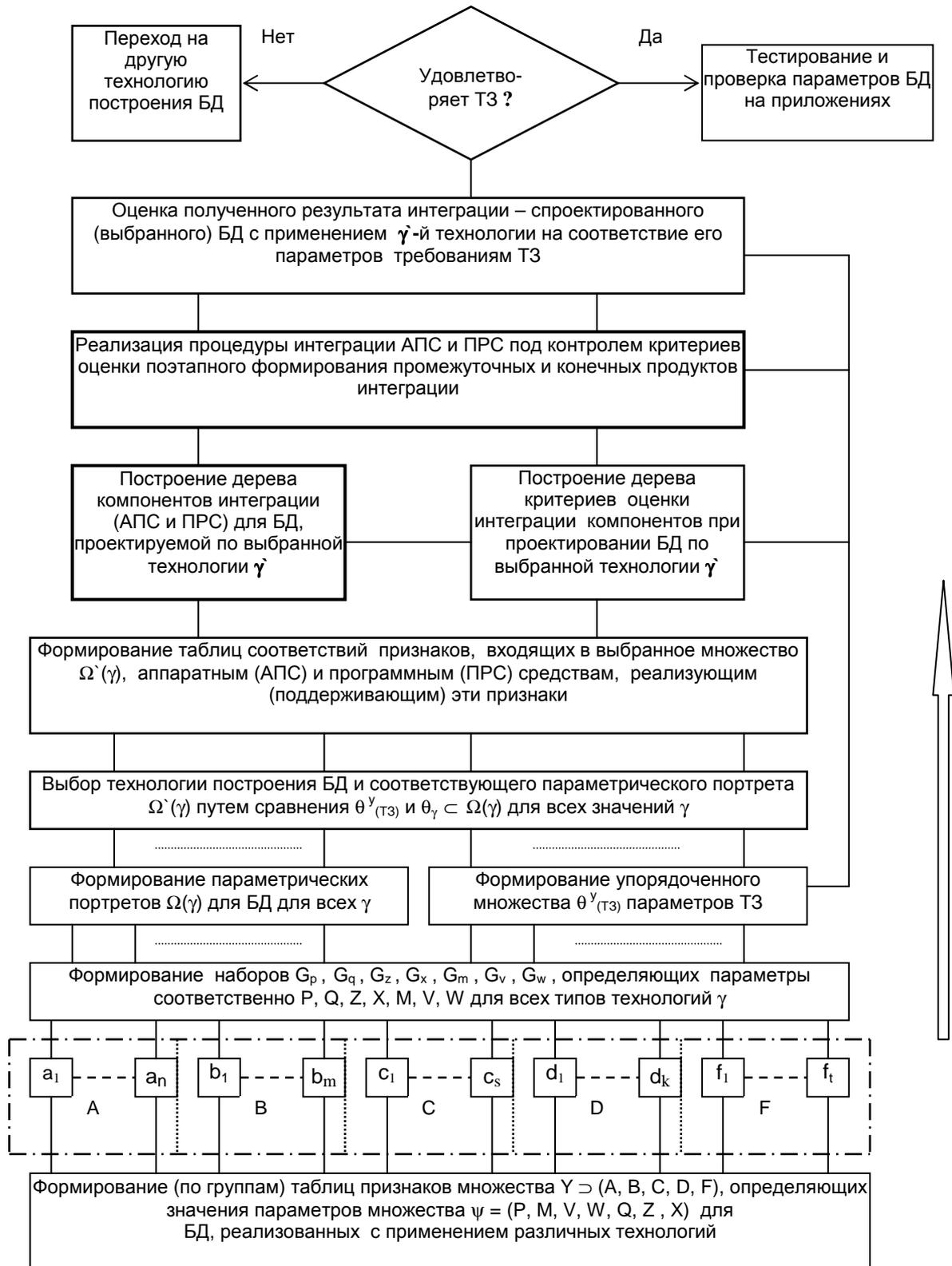


Рис. 1. Укрупненная схема реализации основных положений предложенной концепции разработки

5. Архитектурно-структурная организация и параметры сетей памяти, сетевой коммуникации как физической среды для погружения БД.

6. Информация о языках программирования и инструментальных средствах, используемых для создания приложений при работе с БД.

7. Сформулированные требования к разрабатываемой (выбираемой) БД в соответствии с существующими ГОСТами.

8. Множество Y доминирующих признаков (по группам), определяющих параметры БД для различных технологий их разработки и применения.

9. Таблицы соответствия доминирующих признаков наборам аппаратных и программных средств, реализующих (поддерживающих) эти признаки.

10. Методы и алгоритмы построения и использования дерева интегрируемых объектов и дерева критериев для каждой разновидности технологии разработки БД.

11. Информация о фирмах – производителях выбираемого объекта и др.

12. Набор ограничений различного характера, препятствующих разработке или приобретению выбранной БД (ограниченное финансирование, таможенные пошлины, акценты научно-технической политики ведомства и др.).

13. Другая информация, специфичная для конкретной технологии разработки БД.

Однако, это вовсе не означает, что необходимо всегда работать со всей указанной совокупностью данных. Иногда те или иные решения в сложной стратегии проектирования (выбора) оказываются вполне очевидными. Например, если выбран объектный подход к проектированию ИС в целом и выбран объектный язык программирования, то архитектура распределенной объектно-ориентированной БД [11] может рассматриваться по отношению к другим типам БД как приоритетная.

В качестве инструментального средства может быть применена АСИП, построенная по аналогии с системой, которая описана в [22], либо другая система, которая может быть разработана с использованием принципов многоагентной технологии [23, 24].

Выводы

Построение современных АСИП предполагает прежде всего применение распределенных БД, которые достаточно хорошо зарекомендовали себя в коммерческих системах, поскольку по сравнению с другими типами БД они обладают рядом преимуществ, такими, как:

- возможность размещения составляющих частей задач на различных компьютерах, связанных в сеть, что обеспечивает прием удаленных запросов;
- возможность расширения БД путем простого добавления в сеть дополнительных машин;
- более привлекательные характеристики "цена/производительность", благодаря применению современных прогрессивных сетевых технологий;
- возможность параллельной обработки запросов к системе БД, создание распределенных АСИП;
- возможность построения АСИП в масштабах от Intranet до Internet;
- возможность создания АСИП повышенной надежности и др.

Объектно-ориентированные БД (ООБД), несмотря на ряд проблем, которые возникают при их проектировании (поддержание разнородной multi-media информации, интеграция неоднородных БД, поддержание различных представлений ООБД и т.д.), заняли достойное место среди других разновидностей распределенных БД. Уже несколько лет они успешно применяются (O2, ORION, GemStone, Iris и др.) в системах различного назначения.

К сожалению, в настоящее время отсутствует строгая теория проектирования (выбора) распределенных и тем более объектно-ориентированных БД. Разработка эффективных и корректных

преобразований SQL-запросов является трудной задачей, надежные метрики оценок иллюзорны, задача построения расширяемой архитектуры перебора в значительной степени не решена.

Поэтому предложенные в данной работе основные концептуальные положения подхода к созданию (выбору) таких систем, сформированный набор информационного обеспечения, а также предложенные методы и средства для реализации этого подхода имеют большое теоретическое и практическое значение, несмотря на то, что эти положения формировались не столько по результатам анализа собственного опыта разработок, сколько на основе изучения и анализа результатов исследований специалистов в области создания и применения БД.

Принципиальным отличием предложенной концепции проектирования БД от известных является то, что она основывается на анализе и оценке качественных и количественных признаков, определяемых применяемыми различными технологиями разработки БД, на формировании их параметрических типовых портретов, сравнении этих портретов с параметрическим портретом технического задания с целью определения (с точки зрения удовлетворения требованиям ТЗ) приоритетной технологии построения БД и усеченного набора её доминирующих признаков, а также на методологии интеграции аппаратных и программных средств, соответствующих этим признакам и определяющих их значения. Предложенная концепция ориентирована на использование автоматизированных инструментальных средств проектирования, которые также могут быть созданы с применением новых технологий разработки, что, в принципе, позволит сократить затраты и время, необходимое для проектирования, обеспечив при этом оптимальный вариант построения БД.

СПИСОК ЛИТЕРАТУРЫ

1. Гринев М. Системы управления полуструктурированными данными // Открытые системы. – 1999. – № 5 – 6. – С. 31 – 41.
2. Буров К. Обнаружение знаний в хранилищах данных // Открытые системы. – 1999. – № 5 – 6. – С. 23 – 40.
3. Дубова Н. Устройство и назначение хранилищ данных // Открытые системы. – 1998. – № 4 – 5. – С. 50 – 54.
4. Шапот М. Интеллектуальный анализ данных в системах поддержки принятия решений // Открытые системы. – 1998. – № 1. – С. 30 – 35.
5. Киселев М, Соломатин Е. Средства добычи знаний в бизнесе и финансах // Открытые системы. – 1997. – № 4. – С. 41 – 44.
6. Решение проблемы комплексного оперативного анализа информации хранилищ данных / С.Д. Коровкин, И.А. Левенец, И.Д. Ратманова, В.А. Старых, Л.А. Щавелев // СУБД. – 1997. – № 5 – 6. – С. 47 – 51.
7. Edelstein Н. Интеллектуальные средства анализа, интерпретации и представления данных в информационных хранилищах // CWM. – 1996. – №16. – С. 32 – 35.
8. Кузнецов С. Тенденции в мире систем управления базами данных. <http://citforum.primorye.ru/database/kbd96/48.shtml>.
9. Тамер Оззу М., Валдуриз П. Распределенные и параллельные системы баз данных // Системы управления базами данных. - 1996. – № 4. – С. 45 – 57.
10. Разработка интеллектуальных систем поиска информации в распределенных базах неоднородных ресурсов. <http://www.tsure.ru:8100/University/Faculties/Fib/bit/rus/sem2001/50.htm>.
11. Андреев А.М., Березкин Д.В., Кантонистов Ю.А. Выбор СУБД для построения информационных систем корпоративного уровня на основе объектной парадигмы. http://www.citforum.ru/database/articles/subd_cis.shtml.
12. Развитие архитектур баз данных. <http://www.macworldexpo.ru/os/1995/02/4.htm>.
13. Зиндер Е.З. Проектирование баз данных: новые требования, новые подходы. <http://www.citforum.ru/database/kbd96/41.shtml>.
14. Русин А.О. Распределенная, объектно-ориентированная система управления базами данных с сетевым принципом организации. <http://www.greenleaf.ru/freeware/articles/report2.html>.
15. Шуткин Л. О применении теории паттернов в компьютерных системах. <http://www.osp.ru/os/1995/06/31.htm> - part_1.
16. Цимбал А. Сравнительный анализ технологий CORBA и COM. <http://www.interface.ru/borland/corbacom.htm>.
17. Андреев А.М., Березкин Д.В., Кантонистов Ю.А., Мальцев С.А. WEB-интерфейс объектной СУБД. http://www.inteltec.ru/publish/articles/objtech/4kx6_9.shtml.
18. Доступ к базам данных на стороне клиента. Java-технология. <http://www.webclub.ru/materials/postgres95/1.html>.
19. Яковлев Ю.С. О проблеме интеграции аппаратных и программных платформ при создании компьютерных систем // Кибернетика и системный анализ. – 2000. – № 5. – С.138 –150.
20. Яковлев Ю.С. Критерии оценки интеграции аппаратных и программных платформ компьютерных систем: Сб. Технології створення перспективних комп'ютерних засобів та систем з використанням новітньої елементної бази. – Київ: Ін-т кібернетики ім. В.М Глушкова НАН України, 2000. – С. 4 –15.
21. Зубинский А. Кластеры - о главном // Компьютерное Обозрение. – 2002. – №18 –19. – С. 20 –28.
22. Коваль В.Н., Палагин А.В., Яковлев Ю.С. и др. Интегрированная аналитическая система информационной поддержки (АСИП) в области СВТ: концепция, реализация, перспективы // Проблемы программирования. – 2000. – №1 – 2. – С. 398 – 408.
23. Яковлев Ю.С. Применение многоагентной технологии для разработки компьютерных систем: Сб. Засоби комп'ютерної техніки з віртуальними функціями і нові інформаційні технології. - Київ: ІК НАНУ, 2002. – С. 65 – 71.
24. Яковлев Ю.С. Архитектура агента многоагентной системы для разработки компьютерных систем: Сб. Засоби комп'ютерної техніки з віртуальними функціями і нові інформаційні технології. – Київ: ІК НАНУ, 2002. – С. 59 – 65.