

На правах рукописи



Лукьянчиков Олег Игоревич

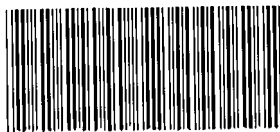
**ТЕХНОЛОГИЯ АГЕНТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ
ДЛЯ ПРОЕКТИРОВАНИЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ
ОБРАБОТКИ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ**

05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

5 АВГ 2015



005571220

Москва 2015

Работа выполнена на кафедре «Управление и моделирование систем» (КБ-3) Федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный университет информационных технологий, радиотехники и электроники».

Научный руководитель:	доктор технических наук, профессор, проректор по научной работе Негосударственного образовательного учреждения высшего образования Московский технологический институт <i>Евгений Витальевич Никульчев</i>
Официальные оппоненты:	доктор технических наук, доцент, старший научный сотрудник лаборатории вычислительных ресурсов Федерального государственного бюджетного учреждения науки Институт вычислительных технологий Сибирского отделения Российской академии наук <i>Владимир Борисович Барахнин</i>
	кандидат технических наук, доцент, доцент департамента программной инженерии Федерального государственного автономного образовательного учреждения высшего профессионального образования Национального исследовательского университета «Высшая школа экономики» <i>Александр Давидович Брейман</i>
Ведущая организация:	Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Рязанский государственный радиотехнический университет»

Защита состоится «29» сентября 2015 г. в 15⁰⁰ на заседании диссертационного совета Д212.131.05 в Московском государственном университете информационных технологий, радиотехники и электроники по адресу 119454, г. Москва, проспект Вернадского, д. 78.

С диссертацией можно ознакомиться в библиотеке МИРЭА.

Автореферат разослан «29» июля 2015г.

Ученый секретарь диссертационного совета,
кандидат технических наук, доцент



Е. Г. Андрианова

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность. Масштабы современных информационных систем постоянно растут, увеличивается количество программных компонентов, и, разумеется, увеличиваются объемы программного кода. В результате увеличивается время на разработку информационных систем и уменьшается качество программных средств.

Тенденции развития информационных систем состоят в обеспечении потребностей обработки огромных объемов данных и обслуживании большого количества клиентов одновременно на разных уголках земного шара. При построении информационных систем распределение вычислений и данных позволяет повысить их производительность и доступность. Для упрощения разработки и интеграции компонентов распределенных систем используется программное обеспечение промежуточного слоя (англ. *middleware*), которое обеспечивает требуемый уровень прозрачности, масштабируемости и открытости взаимодействия. К технологиям и стандартам промежуточного слоя, обеспечивающим удаленный доступ до объектов, относятся: *CORBA, COM, DCOM, COM+, RPC, MOM*.

В распределенных информационных системах широко применяются реляционные БД, а модель приложений является объектной. Такое сочетание значительно затрудняет проектирование и разработку, так как для операций с БД используется язык запросов SQL, никак не связанный с классами и модулями приложений. Обычно при разработке объектно-ориентированных приложений используются различные технологии взаимодействия с реляционными БД, такие как: *ODBC, DAO, OLE DB, BDE, JDBC*. Наиболее предпочтительной является технология объектно-реляционного отображения (*ORM*), основоположником которой является С. В. Эмблер. В связи с различиями объектной и реляционной технологий полное отображение не возможно. Развитием теории совместного использования объектов и БД и практическим приложениям посвящены работы В.П. Иванникова, К. В. Антитина, В. В. Рубанова, В. А. Семенова, С. В. Морозова, и др.

В настоящее время появился ряд архитектурных решений построения систем, ориентированных на предоставление информационных услуг, таких как сервис-ориентированная архитектура и архитектура облачных вычислений. Предоставление облачных услуг не позволяет обеспечивать безопасность доступа к данным и гарантированный уровень обслуживания. Решить данные проблемы позволяет гибридная облачная инфраструктура, в которой часть данных (обработка запросов к которым требует больших вычислительных затрат, а

сами данные не являются конфиденциальными), размещается в общем (публичном) облачном сервисе, а остальные данные — в частном (приватном) облаке или в локальной сетевой инфраструктуре. Гибридные облачные системы имеют сложную архитектуру, которая состоит из большого количества внешних и внутренних узлов, в них используются распределенные БД и системы виртуализации. Поэтому на текущий момент методики управления данными в гибридной облачной инфраструктуре находятся в стадии разработки, решению этих задач посвящены труды множества ученых (*Р. Буйя, Б. Джавади, Б. Ли, Б. Лянг, Б. Сотомейр, Чень Сяофен, Е. В. Никульчев, Е.В. Плужник, Р. С. Монтеро и др.*). Однако отсутствует единое программное решение промежуточного слоя для разработки распределенных систем (в том числе для гибридных облаков), способное объединить возможности удаленного доступа и взаимодействие с БД, а также включать специализированные технологии управления системами виртуализации и возможности переструктурирования данных в процессе эксплуатации систем.

На основе вышеизложенного, актуальным является унификация технологий промежуточного уровня распределенных систем, и разработка единой новой технологии агентно-реляционного отображения (АРО) для проектирования распределенных систем обработки больших объемов данных.

Цель работы: создание технологии агентно-реляционного отображения, предназначенной для разработки распределенных информационных систем обработки больших объемов данных, обеспечивающей гибкость и масштабируемость систем в процессе эксплуатации.

Задачи, решаемые в работе:

1. Обзор и анализ технологий разработки распределенных систем.
2. Формулировка концептуальных положений для построения объектной модели технологии АРО.
3. Разработка методики проектирования распределенных систем с использованием предложенной технологии.
4. Создание прототипа предлагаемой технологии.
5. Экспериментальное исследование технологии для разработки распределенных систем, анализ полученных результатов.

Предметом исследования является автоматизация процесса разработки распределенных информационных систем, ориентированных на обработку больших объемов данных и одновременное обслуживание множества клиентов.

Объектом исследования диссертационной работы являются методы и технологии программирования, которые используются для построения распределенных информационных систем.

Научная новизна. Полученный в работе комплекс теоретических результатов, обобщений и исследований позволил решить научно-техническую задачу создания технологии АРО для проектирования распределенных систем обработки больших объемов данных, обеспечивающей гибкость и масштабируемость в процессе эксплуатации.

Наиболее существенные научные результаты заключены в следующем:

- сформулированы базовые теоретические положения и построена объектная модель агентно-реляционного отображения систем обработки больших объемов распределенных данных;
- разработана методика проектирования распределенных систем на основе технологии агентно-реляционного отображения;
- сформулированы критерии эффективности функционирования систем обработки данных, построенных на основе агентно-реляционного отображения, что позволяет выбрать архитектуру распределенной системы и сервисы обработки данных, оптимизировать структуру хранения и размещения данных.

Достоверность и обоснованность научных положений, результатов, выводов и рекомендаций, приведенных в диссертационной работе, подтверждена результатами исследований эффективности предложенной технологии агентно-реляционного отображения, а также апробацией и обсуждением результатов работы на международных и всероссийских научных конференциях, рецензированием и предварительной экспертизой научных статей, опубликованных в ведущих научных изданиях.

Практическая значимость и внедрение.

Результаты, полученные в диссертационной работе, в виде разработанного прототипа технологии АРО, используются для построения распределенных информационных систем любой сложности. Так как разработка проводилась на С++ с использованием библиотеки QT и с использованием СУБД PostgreSQL, это дает возможность использовать разработки под защищенными операционными системами, такими как MSVC и AstraLinux.

Результаты работы используются для разработки информационных систем в ЗАО «Инженерный центр «ГЕОМИР», которые позволяют автоматизировать и упростить процесс разработки программных средств.

Помимо этого, материалы и разработанные методы диссертационной работы внедрены в учебный процесс МИРЭА и использованы в качестве методологической основы при разработке общеуниверситетских курсов лекций и практических занятий по дисциплинам «Кроссплатформенные технологии», «Современные информационные технологии» и «Технологии распределенной

обработки информации».

Основные положения, выносимые на защиту:

- теоретические положения и объектная модель агентно-реляционного отображения систем обработки больших объемов распределенных данных;
- методика проектирования распределенных систем на основе технологии агентно-реляционного отображения;
- критерии эффективности функционирования систем обработки данных, построенных на основе агентно-реляционного отображения.

Апробация. Результаты диссертации докладывались и обсуждались на двух международных и пяти всероссийских научных конференциях и семинарах.

Международные научные конференции:

- Международная конференция «Инжиниринг & Телекоммуникации — En&T 2014» (МФТИ, Долгопрудный, 2014);
- International Conference on Information Processing and Control Engineering 2015 (МГТУ им. Н.Э. Баумана, Москва, 2015).

Всероссийские научные конференции и семинары:

- Научные чтения памяти К.Э. Циолковского (Калуга, 2013);
- Третий Национальный суперкомпьютерный форум (НСКФ-2014) (ИПС РАН, Переславль-Залесский, 2014);
- Научно-практическая конференция «Актуальные проблемы приборостроения, информатики и социально-экономических наук» (МГУПИ, Москва, 2014);
- Шестая школа-семинар «Задачи системного анализа, управления и обработки информации» (МТИ, Москва, 2015);
- Пятая Всероссийская ежегодная научная конференция памяти академика Ю. В. Бородакия «Современные тенденции развития теории и практики управления в системах специального назначения» (Концерн «СИСТЕПРОМ», Москва, 2015).

Публикации. Результаты диссертации опубликованы в 11 работах, включая 3 статьи в ведущих периодических изданиях, рекомендованных ВАК РФ. В работах, написанных в соавторстве, О.И. Лукьянчикову принадлежат результаты по созданию программно-инструментальных средств проектирования распределенных систем обработки данных, а также агентно-ориентированной технологии проектирования программных систем.

Соответствие паспорту специальности 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных

сетей» по пунктам 3 и 9, так как разработанная технология АРО предназначена для организации взаимодействия программ и программных систем и для организации глобально распределенной обработки данных.

Структура и объем диссертации. Диссертация состоит из введения, 4 глав, заключения, списка использованных источников из 110 наименований, приложений. Общий объем работы 168 с., из них 110 с. — основное содержание, включая 31 рисунков.

ОСНОВНОЕ СОДЕРЖАНИЕ ДИССЕРТАЦИИ

Данная диссертационная работа посвящена новой технологии программирования, которая сочетает в себе современные технологии программирования разработки распределенных систем.

Во введении обосновывается актуальность диссертации, сформулированы цели, задачи, объект и предмет исследования, приведены научная новизна и практическая ценность работы.

В первой главе «Анализ технологий и методов разработки распределенных систем» проведен аналитический обзор основных свойств, которым должна удовлетворять распределенная система, и рассмотрены различные разновидности данных систем, в том числе и гибридная облачная система. Данные системы имеют сложную архитектуру, которая состоит из большого количества внешних и внутренних клиентов, в них используются распределенные БД и системы виртуализации. Поэтому на текущий момент не определена наилучшая технология для разработки приложений для гибридной облачной инфраструктуры, и для этих целей используется совместно множество разнообразных технологий.

Для анализа технологий проектирования и разработки распределенных систем рассмотрены: CORBA, COM, DCOM, COM+, RPC; MOM. Анализ общих свойств и возможностей рассмотренных технологий, позволил выявить обобщенные свойства и провести унификацию, после чего стало возможным отнестись к использованию данных технологий удаленного доступа к парадигме агентно-ориентированного программирования, предложенной *Й. Шохелом* в 1990 годах.

Помимо технологий удаленного доступа в первой главе рассмотрены распределенные БД, проанализированы и структурированы технологии взаимодействия с БД, такие как: *ODBC, DAO, OLE DB, BDE, JDBC*. Однако данные технологии являются неудобными для работы с распределенными БД, так как разработчику необходимо оперировать SQL запросами, при этом контролируя расположение данных и реализуя поддержание ссылочной целостности данных. В качестве предпочтительной технологии предложено использовать технологию

объектно-реляционного отображения (*ORM*), основоположником которой является *С. В. Эмблер*. При *ORM* каждой реляционной таблице БД ставится в соответствие объектно-ориентированный класс приложения, объекты которого содержат возможности для выполнения операций с БД.

В результате предложено совместное использование технологий удаленного доступа и технологии *ORM* их объединение с целью создания универсальной технологии разработки распределенных систем обработки больших объемов данных.

Во **второй** главе «Агентно-реляционное отображение» описаны концептуальные положения предлагаемой технологии и описана методология проектирования систем при ее использовании.

Введено понятие *агентно-реляционного отображения* (*АРО*), под которым подразумевается отображение реляционной модели в объектную модель, экземпляры которой имеют возможность удаленного взаимодействия между собой. Технология *АРО* заключается в том, чтобы предоставить объектам *АРО* возможности по выполнению операций выборки и манипулирования данными, как к таблицам реляционной БД, так и к удаленным объектам *АРО*. В основе технологии, реализующей *АРО*, лежит совместное использование удаленного доступа и концепции *ORM*, а также дополнительные возможности, сформулированные в следующих трех теоретических положениях.

Во-первых, это набор базовых операций, в которые входят выборка и 3 метода манипулирования данными: добавление, удаление, изменение. Данные операции обеспечивают работу с реляционными БД и присутствуют практически во всех реализациях технологии *ORM*, соответственно, при объединении технологий *ORM* и удаленного доступа данные операции должны выполнять помимо реляционных запросов к БД, команды к удаленным объектам, а именно:

- выборка (*select*) — получение данных удаленного объекта;
- изменение (*update*) – установка данных удаленного объекта;
- добавление (*insert*) – добавление в удаленный контейнер из объектов;
- удаление (*delete*) – удаление из удаленного контейнера объекта.

Данные методы позволяют обеспечить синхронизацию данных между всеми удаленными объектами, для обеспечения чего предпочтительным является метод синхронизации через транзакции, потому что данным способом обеспечивается синхронизация данных в реляционных СУБД и поддерживается согласованность данных.

Поэтому **второй** дополнительной возможностью при объединении данных технологий является наличие механизма транзакций для синхронизации дан-

ных. Для реализации механизма транзакций необходимы две операции при удаленном взаимодействии: зафиксировать транзакцию (commit) и откатить транзакцию (rollback). Объект, вызывающий операцию, становится инициатором и выполняет сбор подтверждения о выполнении операции от всех прочих объектов на всех приложениях и на всех вычислительных узлах и фиксацию или откат транзакции по собранным результатам.

В-третьих, для реализации механизма транзакции необходимо знать информацию обо всех удаленных объектах. Таким образом, каждое приложение, использующее данный механизм, должно иметь «карту» всех объектов системы, содержащую следующую информацию о расположении объекта: название объекта (в качестве чего может быть использовано название таблицы); приложение (название и PID номер приложения, уникально идентифицирующий приложение на узле); вычислительный узел (название вычислительного узла и IP-адрес вычислительного узла в сети).

Таким образом, для формирования «карты» объектов каждое приложение должно выполнять регистрацию своих объектов в других приложениях. Операция регистрации должна выполняться при запуске каждого приложения, использующего технологию АРО и обеспечивать отмену регистрации при завершении такого приложения.

Данные теоретические положения позволяют объекту, помимо выполнения запросов к БД, обеспечить удаленное взаимодействие с подобными ему объектами принципами и командами, которыми осуществляется работа с реляционной БД.

На основе теоретических принципов технологии АРО построена объектная модель в нотации языка UML (рисунок 1), в которой:

- классы «Карта объектов» и «Расположение объекта» обеспечивают карту удаленных объектов;
- класс «Параметры сущности» содержит метаданные сущности АРО;
- классы «Базовая сущность» представляет собой сущность АРО;
- каждая сущность содержит объект «Транзакция», обеспечивающий синхронизацию данных между удаленными узлами и БД;
- за всеми сущностями в приложении следит объект класса «Уведомитель», который также отвечает за отправку команд и за получение их от других удаленных объектов сущностей.

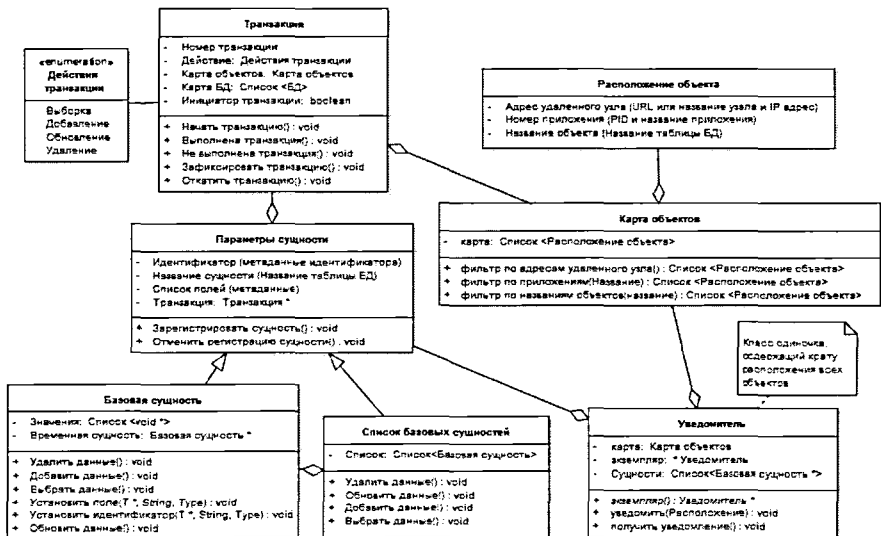


Рисунок 1– Объектная модель технологии агентно-реляционного отображения в нотации языка UML

Для технологии предложена методика проектирования систем с применением АРО, включающая в себя поэтапное формирование следующих моделей.

1. Концептуальная модель. Модель предметной области, в которой определяются основные сущности системы и связи между ними. В ходе данного этапа строится диаграмма «сущность-связь», предпочтительнее в нотации Чена (пример — см. рисунок 2).
2. Логическая модель БД. Создается схема реляционной БД на основе концептуальной модели данных (пример — см. рисунок 2). Предпочтительным способом построения объектной модели является построение диаграммы классов в нотациях языка UML.
3. Объектные модели приложений. На основе концептуальной модели данных, строится объектная модель в виде диаграммы классов в нотациях языка UML. Однако при этом учитывается объектно-реляционное отображение от логической модели данных (пример — см. рисунок 2).

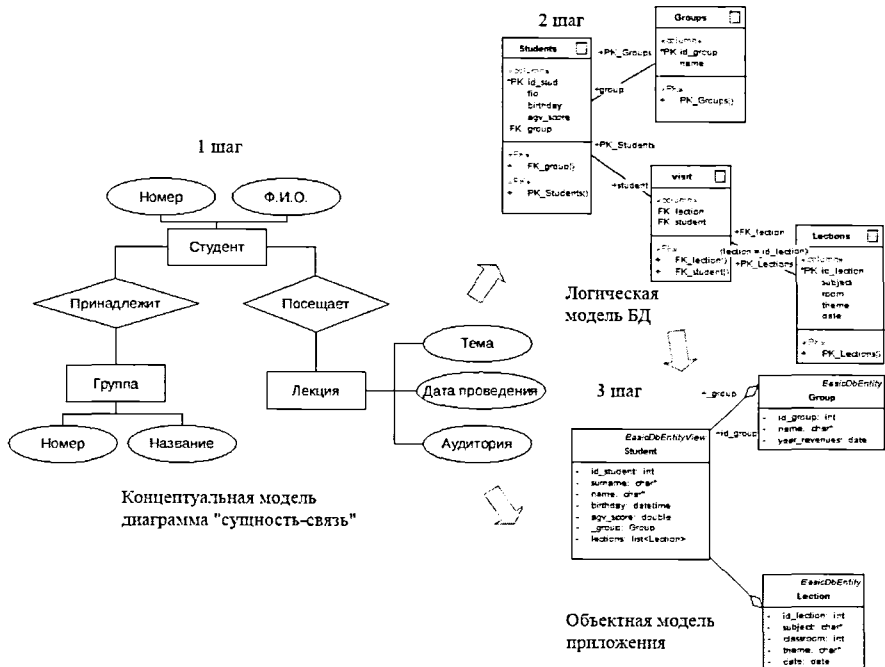


Рисунок 2 – Примеры диаграмм при проектировании

4. Физическая модель БД. Реализация БД на конкретной СУБД.
5. Дополнительные диаграммы языка UML, для приложений. С помощью диаграмм UML в полном объеме описывается система, в том числе и взаимодействие объектов.
6. Физическая реализация приложений. Непосредственная реализация приложений на объектно-ориентированном языке программирования, с использованием технологии АРО.

В третьей главе «Реализация технологии агентно-реляционного отображения» описаны основные возможности технологии, которые упрощают и автоматизируют разработку приложений систем, а также позволяющие расширить функциональность системы, гибкость и масштабируемость:

- скрытие всей работы с SQL, выполняя самостоятельно генерацию всех необходимых запросов;
- предоставление функций для удаленного взаимодействия с другими подобными ему объектами;
- предоставление возможности синхронизации всех объектов в программном комплексе.

После этих трех шагов разработчик может создать объект, которому доступны функции выборки, добавления, обновления и удаления, как в БД, так и на любом удаленном вычислительном узле.

Предложенная реализация технологии АРО выполнена на языке С++ с использованием библиотеки Qt и взаимодействует с СУБД «PostgreSQL». Выбранные языковые средства и библиотеки для разработки позволяют использовать наработки под российские защищенные операционные системы, такие как МСВС и AstraLinux.

Предложенная реализация технологии АРО, включает в себя библиотеку, названную «ArLib», которая непосредственно реализует АРО, и службу «ArNotifyService», обеспечивающую обмен данными между приложениями, использующими библиотеку «ArLib» (принцип обмена – рисунок 3).

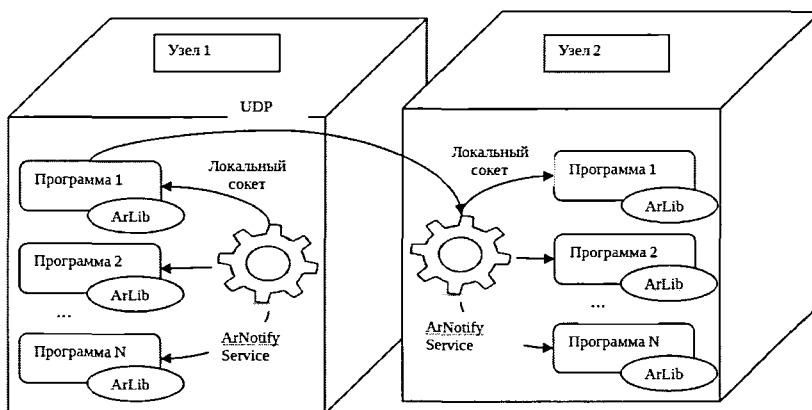


Рисунок 3 - Передача сообщений через службу «ArNotifyService»

Основным классом библиотеки «ArLib» является «ArBasicEntity», наследуясь от которого, объекты получают возможности выполнять запросы в БД и взаимодействовать с другими объектами. Разработана методика использования технология АРО для разработчика программного обеспечения:

1. Создать наследника от класса «ArBasicEntity» и в конструкторе указать 2 параметра: название сущности и название таблицы в БД.
2. В конструкторе объекта нужно указать соответствие атрибутов объектов к полям сущности БД, используя функции «void setField» и «void setIdentifier». Например:

```
setIdentifierField(&_id,"id_stud",("Номер студента",QVariant::Int);
setField(&_fio,"fio",("Фамилия Имя Отчество",QVariant::String);
setField(&_avgScore,"average_score",("Средний балл",QVariant::Double);
```

3. Полученный объект необходимо зарегистрировать в системе, для этого в функции «main» необходимо вызвать метод «void RegisterMetaType<T>», например: *ArEntityParameters::RegisterMetaType<Student>()*;

После этого объект полностью готов к использованию и можно выполнять все стандартные операции: выборку из БД (функцией «bool selectData»); добавление новой записи в БД (функцией «bool insertData»); изменение записи в БД (функцией «bool updateData»); удаление записи из БД (функцией «bool deleteData»). Данные методы принимают в качестве параметров список настроек подключения к БД и список удаленных объектов. Выполнение операций данных методов производится в отдельных транзакциях, которые фиксируются только при успешном выполнении во всех указанных местах.

Второй важный класс библиотеки «ArLib» — «ArNotifier», который обеспечивает взаимодействие между объектами, созданными на основе класса «DbBasicEntity». Объект класса «ArNotifier» является одиночкой (англ. *Singleton*), то есть объект этого класса может быть только один. Объект «ArNotifier» осуществляет непосредственную передачу и прием сообщений, содержащих информацию с сериализованным объектом, с информацией «от кого и кому» и с названием операции. Объект «ArNotifier» получает сообщения от службы «ArNotifyService»; фильтрует не относящиеся к данному приложению сообщения; десериализует объект и по уникальному идентификатору объекта дает команду на выполнения операции, полученной в сообщении; содержит «карту» всех объектов системы; выполняет регистрацию удаленных объектов и отмену регистрации.

В четвертой главе «Анализ эффективности применения технологии агентно-реляционного отображения при построении систем» описаны исследования по оценке эффективности технологии АРО при построении различных классов информационных систем.

Для оценки эффективности использования технологии АРО было произведено сравнение двух классов «Students» (Таблица 1): с использованием технологии АРО и без использования технологии АРО, с реализацией возможностей.

Таблица 1 - Оценка эффективности использования технологии АРО

Метрики	С испол. АРО	Без испол. АРО	Процентное соотношение, %
Общее количество строк в коде	86	241	35,6
Общее количество операторов в коде	44	147	29,9
Процент ветвящихся выражений	0%	11,6%	0
Общее количество методов в классе	9	18	50
Среднее количество выражений на метод	2	5,3	37,7
Максимальная сложность	1	5	20
Максимальная глубина блоков	1	3	33,3
Средняя глубина блоков	0,68	1,14	59,6
Средняя сложность	1	1,94	51,5

Данные результаты получены с помощью ПО «SourceMonitor». По результатам оценки метриками, оценивающими количество программного кода (общее количество строк в коде, общее количество операторов в коде, среднее количество выражений на метод) эффективность использования технологии около 30%, то есть программисту в среднем требуется писать в 3 раза меньше программного кода. По результатам оценки метриками, оценивающих сложность написанного программного кода (общее количество методов в классе, максимальная глубина блоков, средняя глубина блоков, средняя сложность) эффективность использования технологии около 50%, то есть программный код является более читабельным, проще для понимания, а значит, его проще будет сопровождать и дописывать.

Наиболее предпочтительным для оценки производительности технологии АРО являются распределенные системы с реляционными БД, потому что в данных системах применяются все возможности технологии. В данных системах присутствует множество клиентских приложений, которые имеют возможность обмениваться удаленно данными между собой, и распределенная БД, как показано на рисунке 4 (где N – количество клиентов в системе, а M – количество БД в системе). В данных системах предпочтительно оперативные данные хранить на клиентах, а долгосрочные данные, которые редко востребованы, хранятся в реляционных БД. При данной архитектуре клиент, который начал операцию, выполняет синхронизацию данных.

Другой интересной для рассмотрения системой является сервис-ориентированная архитектура (рисунок 4).

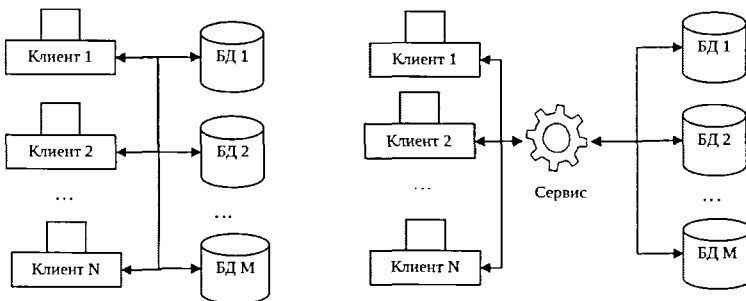


Рисунок 4 – Распределенная система с реляционными БД и сервис-ориентированная архитектура

При данном построении системы имеется централизованный сервис, который будет отвечать за синхронизацию данных и выступать в качестве инициатора операций. Поэтому исследования эффективности направлены на сравнение времени выполнения операций в этих двух разных видах систем.

В качестве критерия эффективности предложен показатель времени выполнения операций при изменяющемся количестве клиентов.

Среднюю оценку производительности распределенной системы с распределенными БД при использовании технологии АРО, можно представить в виде:

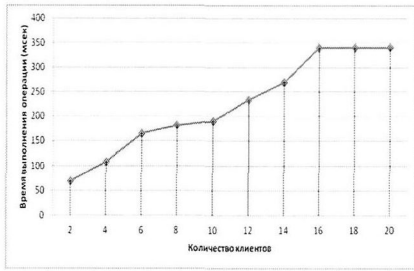
$$\bar{T} = \bar{t}_s + \bar{t}_g + 2 \sum_{i=1}^{N+M} (\bar{t}_{z_i}) + \max \left(\max_{i=1..N} (k\bar{T}_{o_i} + \bar{t}_{a_i}), \max_{i=1..M} (\bar{T}_{db_i} + \bar{t}_{a_i}) \right) + \bar{t}_c, \quad (1)$$

для сервис-ориентированной архитектуры

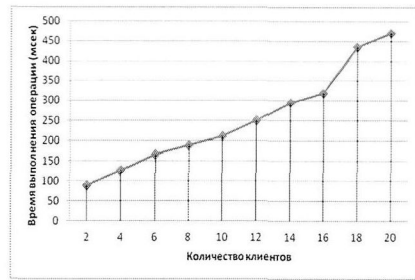
$$\bar{T} = 2\bar{t}_s + \bar{t}_g + \bar{t}_z + 2 \sum_{i=1}^{N+M} (\bar{t}_{z_i}) + \max \left(\max_{i=1..N} (k\bar{T}_{o_i} + \bar{t}_{a_i}), \max_{i=1..M} (\bar{T}_{db_i} + \bar{t}_{a_i}) \right) + \bar{t}_c. \quad (2)$$

Здесь \bar{T} – средняя оценка времени выполнения одной операции; \bar{t}_z – среднее время отправки заявки на выполнение операции; \bar{t}_s – среднее время сериализации объекта; \bar{t}_g – среднее время генерации SQL запроса; \bar{T}_o – среднее время выполнения операции удаленным объектом; k – количество удаленных объектов на узле; \bar{t}_a – среднее время ответа с результатом выполнения операции; \bar{T}_{db} – среднее время выполнения операции на СУБД; \bar{t}_c – среднее время фиксации операции.

Данная оценка не включает в себя загрузку узлов, вероятности неудачных выполнения операций и прочие факторы. Поэтому экспериментально получены результаты оценки производительности для операции обновления, показанные на рисунке 5.



Распределенная система с реляционными БД



Сервис – ориентированная система

Рисунок 5 – Эффективность выполнения операций в зависимости от количества клиентов

По результатам видно, что увеличение количества клиентов и БД влечет за собой увеличение времени выполнения операции, то есть эффективность системы уменьшается. Однако также можно отметить, что за счет асинхронности выполнения операций время выполнения при большом количестве клиентов сильно не будет увеличиваться, а вся нагрузка будет ложиться только на клиента инициатора. Для сервис-ориентированной архитектуры в среднем операции выполнялись на 20 мсек. дольше, чем в прошлом случае. Это вызвано тем, что имеются задержки на передачу дополнительной команды сервису, получив которую он выполняет операции подобные предыдущим экспериментам. Однако в данной архитектуре выявлен узел, выполняющий синхронизацию, то при увеличении производительности данного узла, уменьшится время выполнения операций.

Также в четвертой главе рассмотрено использование технологии АРО при построении систем в гибридной облачной инфраструктуре. Технология АРО использована для разработки принципов управления ресурсами и потоками данных в гибридной облачной инфраструктуре. Для имитации гибридной облачной инфраструктуры собран экспериментальный стенд, включающий два сервера и систему коммутации. Сервер, имитирующий публичное облако имеет больше ресурсов, чем сервер, имитирующий приватное облако. Каждый из серверов представляет собой ферму с виртуальными машинами, управляемыми гипервизорами, дающими возможность оперативного (динамического) управления вычислительными ресурсами сервера (процессорной мощностью). Каждая виртуальная машина развернута на сервере, получает фиксированный объем процессорной мощности в течение некоторой единичной операции, что позволяет избежать нулевых значений ресурсов при значительных нагрузках других виртуальных машин. В соответствии с непредсказуемыми значениями

скачков нагрузки ресурсы должны быть разделены между всеми виртуальными машинами в зависимости от требований качества QoS (*Quality of Service*) для достижения производительности в зависимости от выделенных приоритетов.

Система коммутации представляет собой коммутаторы и маршрутизаторы Cisco, позволяющие динамически управлять каналами связи. В том числе динамически распределять приоритеты по типам трафика.

Существуют ряд исследований *Л. Вана, А. Колмана, Е. В. Никульчева, Е.В. Плужника* по созданию систем управления на основе системы нелинейных дифференциальных уравнений с управлением:

$$\frac{dy}{dt} = f(y, u, t), \quad (3)$$

где $y = (y_1, \dots, y_n)^T \in Y \subseteq \mathbf{R}^n$ — вектор контролируемых параметров; $u = (u_1, \dots, u_m)^T \in U \subseteq \mathbf{R}^m$ — управление; t — время; f — идентифицируемая вектор-функция.

В качестве контролируемого параметрами используются соотношения производительности процессов в зависимости от заданных приоритетов. Подобные параметры были предложены как перспективные подходы достижению QoS при непредсказуемых условиях нагрузки в работах *Х. Чжоу, С. Лу, Й. Люю*. Цель управления состоит в распределении ресурсов таким образом, чтобы текущее соотношение производительности соответствовало заданным приоритетным соотношениям в условиях заданных ограничений.

На основании проведенных исследований на экспериментальном стенде, определен круг ситуаций, при которых используемая схема управления не может на данной архитектуре обеспечить гарантированное качество обслуживания. К таким ситуациям можно отнести: добавление нового сервиса в программное обеспечение, ставшего пользоваться большой популярностью, большой рост объема данных, потребовавший увеличения числа виртуальных машин, дополнительных кэш-серверов для гарантированной доставки контента географически удаленным пользователям и т. п. В этих случаях требуется переструктурирование системы, что, в свою очередь, требует переписывание всего программного обеспечения.

Адаптивность к изменению структуры данных и методам их обработки разработанной в ходе диссертационного исследования технологии АРО, позволяет модифицировать код программного обеспечения на 5%.

На основании проведенных экспериментальных исследований и результатов внедрения сделан вывод об эффективности разработанной технологии для повышения качества проектирования распределенных информационных систем.

В результате исследований, изложенных в четвертой главе, сделан вывод о том, что систем обработки распределенных данных, построенных на основе агентно-реляционного отображения, позволяют создать адаптивные и масштабируемые системы. Для конкретных приложений технология АРО позволяет обеспечивать выбор эффективной архитектуры, условий эксплуатации, сервисов обработки данных, структуру хранения и размещения данных с минимальными изменениями в программном коде приложений.

В заключение диссертации приведены основные результаты и выводы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

1. Проведен обзор и анализ технологий разработки распределенных систем, в результате выявлены обобщенные свойства и проведена унификация технологий, что дало возможность отнести использование данных технологий к парадигме агентно-ориентированного программирования.
2. Описаны теоретические основы и построена объектная модель новой технологии программирования распределенных систем, названной агентно-реляционное отображение, которая позволяет выполнить отображение реляционной модели в объектную модель, экземпляры объектов которой имеют возможность удаленного взаимодействия между собой.
3. Разработана методика проектирования систем с использованием предложенной технологии, которая идет от проектирования реляционной БД и затем отражается и перерабатывается в объектную модель приложения.
4. Создан прототип программного обеспечения промежуточного слоя, реализующий технологию АРО для разработки распределенных систем обработки больших объемов данных, который возможно использовать под российские защищенные операционные системы.
5. Результаты экспериментальных исследований и внедрения показали, что разработанная технология позволяет без значительного изменения программного кода менять архитектуру системы, делая ее гибкой и масштабируемой; разрабатывать системы любой сложности, в том числе и в гибридной облачной инфраструктуре.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

В журналах, включенных в список ВАК:

1. Лукьянчиков О.И., Никульчев Е.В., Паяин С.В., Плужник Е.В. Проектирование распределенных информационных систем обработки больших объемов данных в гибридной облачной инфраструктуре // *Прикаспийский журнал: управление и высокие технологии*. 2014. № 4(28). С. 76–87.

2. Никульчев Е.В., Плужник Е.В., Лукьянчиков О.И. Проектирование распределенных информационных систем обработки больших объемов данных в гибридной облачной инфраструктуре // *Вестник Рязанского государственного радиотехнического университета*. 2014. № 50-1. С. 135–138.
3. Лукьянчиков О.И. Технология агентно-реляционного отображения // *Современные проблемы науки и образования*. 2015. № 1; Режим доступа: URL: <http://www.science-education.ru/121-18623>.

В других изданиях:

4. Лукьянчиков О. И. Применение теории управления проектами в наземных комплексах управления полётами космических аппаратов // Труды XLVIII чтений посвященных разработке научного наследия и развитию идей К.Э. Циолковского. Секция «Проблемы ракетной и космической техники» (г. Калуга, 17–19 сентября 2013). Казань: Изд-во Казан. ун-та, 2014. С. 219–230.
5. Плужник Е. В., Никульчев Е.В., Паяин С.В., Лукьянчиков О. И. Проектирование распределенных систем в гибридной облачной инфраструктуре // Международная конференция «Инжиниринг & телекоммуникации – En&T 2014». Тезисы докладов. М.: Долгопрудный: МФТИ, 2014. С. 212–214.
6. Lukyanchikov O., Pluzhnik E., Payain S.V., Nikulchev E. Using Object-Relational Mapping to Create the Distributed Databases in a Hybrid Cloud Infrastructure // (IJACSA) *International Journal of Advanced Computer Science and Applications*. 2014. Vol. 5. No. 12. P. 61–64.
7. Лукьянчиков О. И. Технология объектно-реляционного отображения реляционных данных при агентно-ориентированном программировании // Задачи системного анализа, управления и обработки информации. Межвузовский сборник научных трудов. Выпуск 5. М.: Изд. МТИ, 2015. С. 85–96.
8. Pluzhnik E., Lukyanchikov O., Nikulchev E., Payain S. Designing Applications with Distributed Databases in a Hybrid Cloud // 2015 the 5th international workshop on Computer Science and Engineering (WCSE 2015). Moscow, 15–17 apr. 2015. P. 318–321.
9. Лось В. П., Лукьянчиков О. И. Повышение уровня безопасности за счет перестроения архитектуры систем // *Информационное противодействие угрозам терроризма*. 2015. № 24. С. 18–24.
10. Nikulchev E., Lukyanchikov O., Pluzhnik E., Virukov D., Payain S. Experimental study of the Cloud Architecture Selection for Effective Big Data Processing // (IJACSA) *International Journal of Advanced Computer Science and Applications*. 2015. Vol. 6. No. 6. P 22-26.
11. Лось В. П., Лукьянчиков О. И. Профилирование распределенных БД // *Информационное противодействие угрозам терроризма*. 2015. № 25. С. 138–141.

Handwritten mark or signature in the top right corner.

Подписано в печать 07.07.2015. Формат 60×84 1/16. Усл.печ.л 1.

Отпечатано в типографии «Вишневый пирог»
115114 Москва, 2-й Кожвнический пер., д.12
Заказ 41349 .