

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.657

Бузюк
Илья Александрович

Проблемы доступа к реляционным базам данных по технологии REST

АВТОРЕФЕРАТ

на соискание академической степени
магистра технических наук

по специальности 1-40 80 05 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Научный руководитель
Лукьянец В.Г.
к.т.н., доцент

Минск 2014

КРАТКОЕ ВВЕДЕНИЕ

Архитектурный стиль представляет собой набор ограничительных условий, применяемых при каком-либо конструировании. Архитектурный стиль программных средств является описанием компонентов, которые могут быть использованы для реализации программной системы. REST (Representational state transfer) представляет собой архитектурный стиль, который можно использовать для создания программных средств, в которых клиенты могут отправлять запросы веб-службам.

Человек по имени Рой Томас Филдинг (Roy Thomas Fielding) в 2000 году первым ввел термин REST в качестве концепции в своей докторской диссертации «Архитектурные стили и проектирование архитектур программных систем, поддерживающих работу в сети» («Architectural Styles and the Design of Network-based Software Architectures»). Он был одним из тех, кто работал над спецификацией, определяющей сегодня функционирование Интернета - протоколом HTTP. Рой Филдинг собрал в одну диссертацию все то, что уже существовало, и изложил, как можно получать максимальную выгоду из уже сформировавшейся архитектуры сети. Филдинг использовал термин REST для описания техники и наилучших практик получения данных через HTTP с целью последующего использования в приложениях.

Архитектура в стиле REST состоит из клиентов и серверов. Клиенты инициируют запросы к серверам, серверы обрабатывают запросы и возвращают подходящие ответы. В общем случае REST является очень простым интерфейсом управления информацией без использования каких-то дополнительных внутренних прослоек. Каждая единица информации однозначно определяется глобальным идентификатором, таким как URL. Каждая URL в свою очередь имеет строго заданный формат.

Простота является основным преимуществом данной архитектуры. Актуальность использования архитектуры REST при построении веб / мобильных приложений заключается в повсеместном использовании RESTful сервисов (сервисов реализованных с использованием принципов REST).

Архитектура REST позволяет резко снизить инвестиции, необходимые для обеспечения сервис-ориентированного доступа к корпоративным ресурсам. Такие известные приложения как Twitter, Instagram, Amazon предоставляют REST API для доступа к своим ресурсам. Для увеличения популярности своего продукта компании предлагают открытые API, которыми могут воспользоваться сторонние разработчики.

Базы данных являются неотъемлемой частью практически всех веб-приложений. Большинство СУБД используют язык SQL, так как он удобен для описания логических подмножеств БД. SQL является, прежде всего, информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Целью диссертационной работы является разработка алгоритмов и программного обеспечения для упрощения процесса получения данных из реляционных БД с помощью архитектуры REST.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Выполнить анализ архитектуры REST.
2. Исследовать эффективность двух подходов при построении сервис-ориентированного ПО - REST и SOAP.
3. Разработать архитектуру и алгоритмы программной системы автоматизирующую процесс получения данных из реляционных БД.
4. Реализовать ПО для получения данных из реляционных БД.
5. Провести экспериментальные исследования разработанной системы.

Объектом исследования являются системы организации доступа к удаленным ресурсам.

Предметом исследования является процесс получения данных из реляционных БД с помощью архитектуры REST.

Основной *гипотезой*, положенной в основу диссертационной работы, является возможность использования REST API для получения данных из реляционных БД основанная на соответствии HTTP методов (GET, POST, PUT, DELETE) и SQL операторов манипуляции данными (SELECT, INSERT, UPDATE, DELETE).

Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

Работа выполнялась в соответствии научно-техническими заданиями и планами работ кафедры «Программное обеспечение информационных технологий», и хозяйственными договорами с предприятиями Республики Беларусь: «Разработать модели, методы, алгоритмы для оценки параметров, повышения надежности и качества функционирования аппаратно-программных средств систем и сетей сложной конфигурации и внедрить в современные обучающие комплексы» (ГБ № 11-2004, № ГР 20111065, научный руководитель НИР – В. В. Бахтизин).

Личный вклад соискателя

Результаты, приведенные в диссертации, получены соискателем лично. Вклад научного руководителя В. Г. Лукьянца, заключается в формулировке целей и задач исследования.

Апробация результатов диссертации

Основные положения диссертационной работы докладывались и обсуждались на 49-ой научной конференции аспирантов, магистрантов и студентов БГУИР «Экономика, управление, информационные технологии» (Минск, Беларусь, 2013); XVIII Международной научно-технической конференции «Современные средства связи» (Минск, Беларусь, 2013); XIX Международной научно-технической конференции «Современные средства связи» (Минск, Беларусь, 2014).

Опубликованность результатов диссертации

По теме диссертации опубликовано 2 печатных работ, в сборниках трудов и материалов международных конференций.

Структура и объем диссертации

Диссертация состоит из введения, общей характеристики работы, трех глав, заключения, списка использованных источников, списка публикаций автора и приложений.

В первой главе представлен анализ предметной области, выявлены основные существующие концепции в рамках тематики исследования, произведен анализ двух подходов при построении сервис-ориентированного ПО - REST и SOAP. Вторая глава посвящена эмпирическому исследованию REST и SOAP веб-сервисов для получения данных из реляционных БД. Представлены результаты экспериментальных исследований эффективности двух подходов при построении сервис-ориентированного ПО. Третья глава посвящена разработке архитектуры ПО и алгоритмов программной системы автоматизирующую процесс получения данных из реляционных БД. Предложена практическая реализация ПО.

Общий объем работы составляет 59 страниц, из которых основного текста – 40 страниц, 18 рисунков на 7 страницах, 7 таблиц на 5 страницах, список использованных источников из 34 наименований на 2 страницах и 3 приложения на 5 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ

Во **введении** определена область и указаны основные направления исследования, показана актуальность темы диссертационной работы, дана краткая характеристика исследуемых вопросов, обозначена практическая ценность работы.

В **первой главе** проведен анализ предметной области, выявлены основные существующие концепции в рамках тематики исследования, произведен анализ двух подходов при построении сервис-ориентированного ПО - REST и SOAP. Проанализированы фундаментальные принципы построения и задачи распределенных систем, изучена топология Интернета. Изучены подходы построения сервис ориентированной архитектуры SOA и REST. Выполнена сравнительная характеристика REST и SOAP.

Распределённая система — система, для которой отношения местоположений элементов (или групп элементов) играют существенную роль с точки зрения функционирования системы, а, следовательно, и с точки зрения анализа и синтеза системы.

В широком смысле REST означает концепцию построения распределённого приложения, при которой компоненты взаимодействуют наподобие взаимодействия клиентов и серверов во Всемирной паутине.

Архитектурный стиль программных средств является описанием компонентов, которые могут быть использованы для реализации программной системы. REST представляет собой архитектурный стиль, который можно использовать для создания программных средств, в которых клиенты (агенты пользователей) могут отправлять запросы службам (конечным точкам)

REST изначально описан в контексте HTTP, но не ограничен этим протоколом. Архитектуры типа REST могут быть основаны на других протоколах прикладного уровня, если они уже реализуют обширный и единый словарь для приложений, основанных на передаче значимых представлений состояний.

Результаты исследований, проведенных в этих направлениях, отражены в работах Р. Т. Филдинга (R. T. Fielding), Т. Д. Бернерса-Ли (T. D. Berners-Lee), Д. Б. Постеля (J. B. Postel), Д. А. Рейнольдса (D. A. Reynolds), Д. Л. Вальдо (D. L. Waldo), А. С. Таненбаума (A. S. Tanenbaum), В. Л. Бройдо и др.

Вторая глава посвящена эмпирическому исследованию REST и SOAP веб-сервисов для получения данных из реляционных БД. Представлены результаты экспериментальных исследований эффективности двух подходов при построении сервис-ориентированного ПО.

Объектом исследования являются REST и SOAP веб-сервисы предоставляющие следующий функционал:

- выборка данных из БД;
- запись данных в БД.

Каждый веб-сервис имеет поддержку GET и POST запросов для выборки и добавления данных соответственно. Данный функционал является критической секцией производительности для большинства приложений. В качестве

исходных данных использовались следующие значения: количество пользователей варьируется от 1 до 1000, время начала работы 1000 мс., количество циклов выполнения для каждого пользователя – 1.

Для исследования эффективности веб-сервисов в рамках данной диссертации получены необходимые данные среднего времени выполнения с учетом среднеквадратического отклонения. Результаты исследования отражают среднее время выполнения сервиса с учетом погрешности стандартного отклонения:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}. \quad (1)$$

где x_i — i -й элемент выборки; n — объём выборки; \bar{x} — среднее арифметическое выборки.

Таблица 1 – Результаты однократного выполнения сервисов

Операция	SOAP [мс]	REST [мс]
Чтение из БД	244.2 ± 2.83	233.0 ± 2.36
Запись в БД	309.7 ± 7.59	292.9 ± 6.5

Таблица 2 – Результаты многократного выполнения сервисов для выборки

Количество операций чтения	SOAP [мс]	REST [мс]
5	934 ± 14.5	823 ± 14.3
10	1423 ± 28.7	1248 ± 27.3
20	2123 ± 58.6	1723 ± 55.6
50	4205 ± 110.3	3923 ± 106.6
100	8498 ± 230.2	7112 ± 225.2
500	16835 ± 260.6	15734 ± 256.7
1000	38434 ± 521.8	35893 ± 519.8

Таблица 3 – Результаты многократного выполнения сервисов для записи

Количество операций записи	SOAP [мс]	REST [мс]
5	1214 ± 40.2	1064 ± 39.2
10	1824 ± 81.5	1574 ± 79.6
20	2736 ± 164.4	2213 ± 162.8
50	5461 ± 351.9	5073 ± 347.1
100	11233 ± 704.5	10231 ± 700.9
500	56834 ± 1512.6	52713 ± 1493.5

Анализ табличных данных показывает, что критическим количеством потоков выполнения операций являются около тысячи для чтения и пятисот для записи соответственно. Производительность работы сервисов резко падает по достижению этих значений. Это значит, что на сервере операции становятся в очередь и производительности не хватает, чтобы обслужить все запросы. Данное утверждение справедливо как для REST, так и для SOAP сервиса.

Тем не менее, эффективность REST подтверждается полученными эмпирическими данными. В данном случае для измерения эффективности сервисов воспользуемся формулой подсчета временной производительности:

$$100 - \frac{t_1 + \sigma_1}{t_2 + \sigma_2} * 100 \quad (2)$$

где t_1 — время выполнения первого сервиса; t_2 — время выполнения второго сервиса; σ_1 — среднее квадратичное временное отклонение первого сервиса; σ_2 — среднее квадратичное временное отклонение второго сервиса.

Таким образом, исходя из результатов исследования были получены следующие данные, которые показывают на сколько REST сервис эффективнее аналогичного SOAP сервиса для получения данных из реляционной БД.

Таблица 4 – Эффективность REST по сравнению с SOAP сервисом

Операция	Производительность [%]
Однократное чтение	4.72
Однократная запись	5.63
Многократное чтение	6.52
Многократная запись	7.09

Третья глава посвящена разработке архитектуры ПО и алгоритмов программной системы автоматизирующую процесс получения данных из реляционных БД. Предложена практическая реализация ПО.

Концепция предоставления REST API основана на схожести HTTP методов (GET, POST, PUT, DELETE) и SQL операторов манипуляции данными (SELECT, INSERT, UPDATE, DELETE) таблица 5.

Таблица 5 – Концепция взаимодействия SQL и HTTP

Операция	SQL	HTTP
Create	INSERT	POST
Read	SELECT	GET
Update	UPDATE	PUT
Delete	DELETE	DELETE

Таким образом если не рассматривать реализации HTTP и SQL подходов, то с абстрактной точки зрения все операции сводятся к чтению, записи, обновлению и удалению данных.

Алгоритм преобразования REST в SQL запрос основан на использовании хеширования, где ключами являются REST запросы а значениями соответственно выступают SQL запросы.

Реализация данного алгоритма проста и основана на использовании хеш-таблицы для хранения пар ключ / значение (REST и SQL запрос соответственно). Важной особенностью данного алгоритма является временная сложность $O(n)$ в худшем случае и $O(1)$ в среднем, что является крайне эффективным с точки зрения производительности.

С архитектурной точки зрения необходимо реализовать программный модуль, состоящий из административной консоли с интуитивно-понятным пользовательским интерфейсом и балансера (RTE) отвечающего за корректную обработку запросов, который будет хоститься вместе с приложением на сервере.

На рисунке 1 представлена модель предоставления REST API для получения данных из реляционных БД.

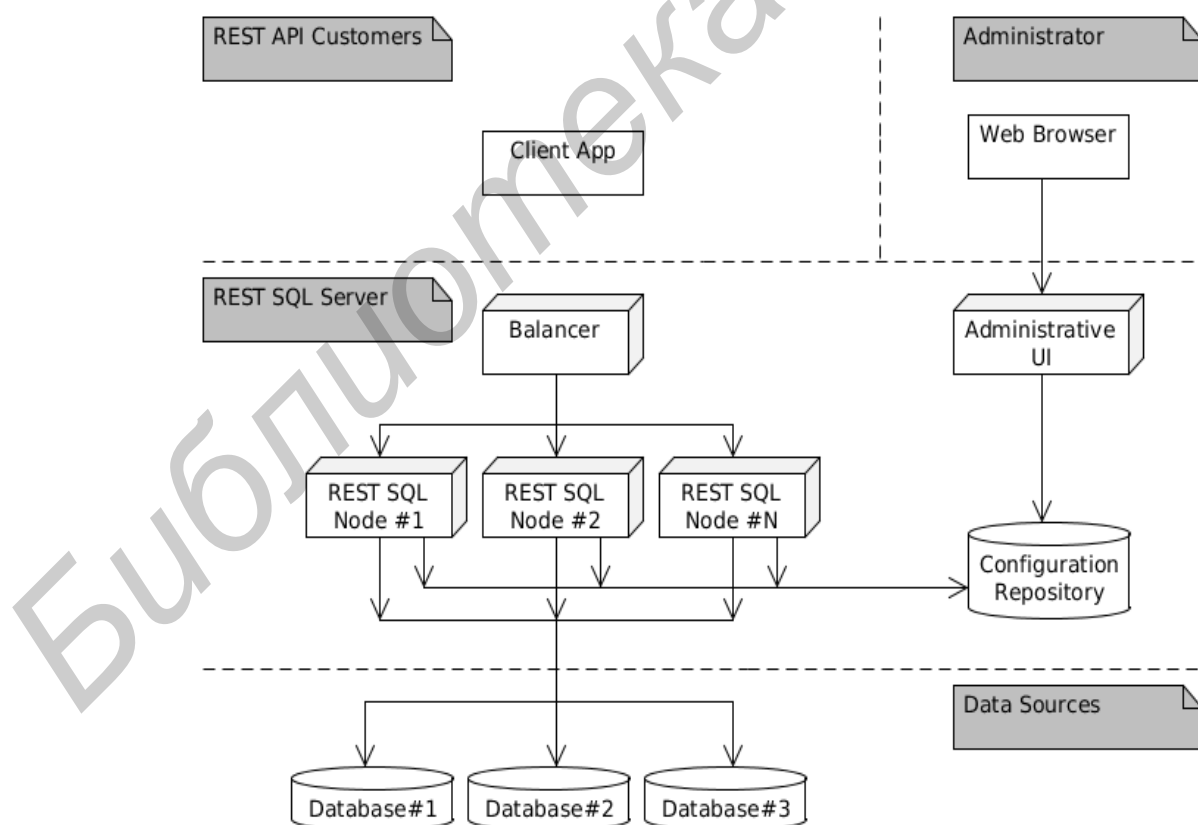


Рисунок 1 – Модель предоставления REST API для получения данных из реляционных БД

Архитектурное решение состоит из трех частей (рисунок 2.2): административного модуля (Administrative UI), балансера (Balancer Real-time engine) и хранилища конфигураций (Configuration Repository).

С помощью наглядного и интуитивно-понятного пользовательского интерфейса административного модуля генерируются REST сервисы на основе SQL запросов. Так же административный модуль отвечает за безопасность, фильтрацию и проверку параметров REST сервиса. Для оптимизации производительности используется пул соединений к базе данных, который так же может быть сконфигурирован в административном модуле. Балансер выступает шлюзом к созданному REST сервису, который отвечает за корректную обработку HTTP запроса (соединение с БД, обработка SQL запроса, возврат запрашиваемых данных в формате xml или json). Хранилище конфигураций необходимо для сохранения ранее созданных ресурсов между запусками серверов.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

1. Исходя из результатов исследования REST сервис имеет значительное преимущество в производительности по сравнению с аналогичным SOAP сервисом для получения данных из реляционной БД. Эффективность использования REST сервиса составляет до 6.52 процентов для многократного чтения и 7.09 для многократной записи соответственно [1].

2. Архитектура REST позволяет резко снизить инвестиции, необходимые для обеспечения сервис-ориентированного доступа к корпоративным ресурса по сравнению с SOAP подходом, который является более тяжеловесным вариантом с точки зрения машинной обработки.

3. REST сервисы просты в реализации поскольку они базируются на хорошо известных Web-протоколах. Простота является основным преимуществом данной архитектуры. Актуальность использования архитектуры REST при построении веб / мобильных приложений заключается в повсеместном использовании RESTful сервисов.

4. Ключевым понятием в REST является наличие ресурсов (источников конкретной информации), каждый из которых определяется глобальным идентификатором. Приложение может взаимодействовать с ресурсами, зная две вещи: идентификатор ресурса и требуемое действие.

5. На основе результатов исследования было разработано готовое CRUD (CREATE, UPDATE, READ, DELETE) решение для работы с персистентными хранилищами данных. Реализация основана на схожести HTTP методов (GET, PUT, POST, DELETE) и SQL операторов манипуляции данными (SELECT, INSERT, UPDATE, DELETE) [1–А].

6. Программное решение предоставления REST API для получения данных из реляционных БД является кроссплатформенным и независимым от конкретной СУБД (Oracle, MySql, PostgreSQL) [2].

Рекомендации по практическому использованию результатов

1. Программное решение можно использовать как простое и эффективное средство для создания REST сервисов на основе SQL запросов. Данный продукт идеально подходит для B2B и B2C решений. Применяя данное программное решение можно значительно упростить организацию доступа к реляционной базе данных приложения и / или предоставить REST API для программного продукта.

2. На данный момент осуществляется поддержка двух самых популярных серверов - JBoss и Tomcat.

3. С помощью наглядного и интуитивно-понятного пользовательского интерфейса административного модуля можно в кратчайшие сроки сгенерировать необходимое количество REST сервисов для получения данных из реляционных БД, не написав при этом ни строчки программного кода, что, по сути, является средством автоматизации создания REST сервисов.

4. Программный продукт, который является результатом диссертации на соискание академической степени магистра технических наук, успешно прошел стадию апробации и тестирования в ИООО «Exadel». Программное решение успешно используется на нескольких внутренних проектах данной компании.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1. Бузюк, И.А. Организация доступа к реляционным базам данных по технологии REST / И.А. Бузюк // Материалы XVIII Междунар. науч.-техн. конф. "Современные средства связи" МИНСК:ВГКС, 15-16 окт. 2013 года. – с. 167.

2. Бузюк, И.А. Новый взгляд на трехуровневую клиент-серверную архитектуру / И.А. Бузюк // Материалы XIX Междунар. науч.-техн. конф. "Современные средства связи" МИНСК:ВГКС, 14-15 окт. 2014 года. – с. 101.