разуют типичные модули ГМЦР: терминальные и прекапиллярные артериолы → капиллярная сеть → посткапиллярные и собирательные венулы (Петренко В.М., 2009). Это соответствует в целом центрифугальному упрощению строения стенок основных звеньев КР. КР в брыжейке двенадцатиперстной кишки (БДК), особенно магистрального типа с краевым анастомозом (вдоль брыжеечного края ДК) между верхними и нижними панкреатодуоденальными артериями и венами (ПДА, ПДВ), напоминает микрорайон ГМЦР в брыжейке тонкой кишки: ветви и притоки ПДА, ПДВ и их анастомозов покрывают головку поджелудочной железы и погружаются в нее, в конечном счете формируют капиллярную сеть. У эмбрионов и плодов человека 6-20 нед КР-БДК состоит из разных микрососудов, толщина и сложность строения их стенок постепенно увеличиваются, начиная от корня БДК. На 6-й нед ПДА имеют строение прекапилляра, ПДВ – капилляра, на 8-й нед они напоминают метартериолы и мелкие посткапиллярные венулы, а на 9-й нед – мелкие терминальные артериолы и первичные собирательные венулы, их сопровождают первичные лимфатические сосуды с эндотелиальными стенками (первичные вены, выключенные из кровотока). Кнутри от контура КР-БДК определяюся прекапилляры и посткапиллярные венулы, сети кровеносных и лимфатических капилляров. У плодов 10-11 нед в составе контура КР-БДК определяются зачатки панкреатодуоденальных лимфоузлов (~ периваскулярные лимфоидные узелки в микрорайонах дефинитивного ГМЦР). Такой общий план строения КР-БДК сохраняется в дальнейшем, хотя возникают дефинитивные варианты его анатомического строения (крайние – магистральная и рассыпная формы), наблюдается «созревание» основных звеньев в связи с увеличением функциональной нагрузки на КР.

Политические науки

«ЧЕРНЫЕ» КОММУНИКАЦИИ – НОВЫЙ ТЕРМИН В ПОЛИТИЧЕСКОЙ НАУКЕ

Оришев А.Б.

Институт бизнеса и дизайна, Москва,
e-mail: orishev71@mail.ru

Термин «черные» коммуникации существует недавно. Более того, он еще не овладел умами миллионов российских политологов, политтехнологов, социологов и прочих специалистов [1]. Им удобней пользоваться привычным словосочетанием «черный» пиар. Стоит ли в таком случае вводить в оборот новое понятие? Ответим на ряд важных вопросов.

Можно ли говорить о «черных коммуникациях»? Для специалистов по связям с общественностью такого понятия просто не существует [3]. Более того, для большинства не существует и понятия «черный» пиар. Дело в том, что если исходить из дословного понимания термина «пиар» – то это технология создания и внедрения образа объекта в массовом сознании как идеального и необходимого. Иногда профессионалы-пиарщики ссылаются на определение генерального секретаря Международной ассоциации PR, профессора Сэма Блэка: «PR – есть искусство и наука достижения гармонии между организацией и общественностью с помощью взаимопонимания, основанного на правде и полной информированности». Т.е. не может быть «черного», «серого», «зеленого» и т.п. пиара, ибо по определению он только «белый» или лишен цветовой гаммы [2].

Если снять «розовые очки», то очень быстро выяснится: черный цвет в коммуникациях есть! Как есть в мире добро и зло. Так и информация может быть правдивой или лживой, исходя из того какие цели ставит коммуникатор: объективно осветить события или манипулировать массами. Более того, любые сведения, даже исключительно положительного характера можно подать в негативном свете. Поэтому скажем прямо и честно – «черные» коммуникации существуют. И они объективная реальность для мира бизнеса и политики.

Список литературы

1. Оришев А.Б., Андропов З.В. Лингвотехнологии как культурно-информационный ресурс в практике различных политических акторов // Экономика и социум. 2013. № 4-3 (9). С. 66-69.
2. Оришев А.Б. Конструирование врага как политическая технология // Международный журнал экспериментального образования. – 2014. – № 3. – С. 156.
3. Оришев А.Б. Публичное выступление: технология ответов на каверзные вопросы // Наука, образование, общество: проблемы и перспективы развития. Сборник научных трудов по материалам научно-практической конференции 28 февраля 2014 г. Ч. 1. – Тамбов: ТРОО «Бизнес-Наука-Общество», 2014. – С. 105-106.

Технические науки

О ПРИМЕНЕНИИ АДАПТИВНОЙ МОДЕЛИ ОРГАНИЗАЦИИ ДАННЫХ В НЕРЕЛЯЦИОННЫХ ХРАНИЛИЩАХ

Васильев А.В.

Московский физико-технический институт (государственный университет), Москва,
e-mail: alexey@phystech.edu

В настоящее время компьютерная индустрия столкнулась с проблемой постоянно растущей сложности программного обеспечения (ПО), с увеличением числа различных протоколов, интерфейсов и используемых в них технологий. Как следствие, общий тренд в индустрии направлен на создание общих библиотек, компонентов и модулей, которые могут быть использованы в различных проектах. Об этом свидетельствует появление огромного количества open-source библиотек, которое непрерывно растет. Использование общего кода позволя-

ет свести число ошибок в таком программном обеспечении к минимуму: их наверняка будет меньше, чем в разработанном внутри компании компоненте. По сути, над тестированием и исправлением ошибок в общих библиотеках трудится все open-source-community.

Отметим, что идея создания универсального ПО реализуется также и внутри компаний, придерживающихся стратегического подхода к построению архитектуры. Конечно, чем более общим будет подход, тем лучше, но и, однако, тем дороже он обойдется на начальном этапе. Поэтому вопрос о необходимости и целесообразности создания общих частей должен рассматриваться с прагматичной точки зрения. Как правило, обобщающие подходы применяются в компаниях, которые производят большое количество продуктов, имеющих типовые составляющие: общую часть в различных проектах, или общую платформу для реализации проектов, или и то и другое.

Что касается проектирования хранилищ данных, то существует множество инструментов для их создания и множество систем управления ими [1]. Однако структура хранилища данных (или просто базы данных) обычно разрабатывается под конкретный проект. Вместе с этим очевидно, что имеет смысл построить общую инфраструктуру для хранения данных с использованием адаптивных моделей хранения данных. В свое время от таких моделей отказывались, так как их показатели были хуже, чем показатели модели данных, созданной для конкретного проекта. Но ухудшение это было связано с особенностями реляционных хранилищ данных – по сути, единственных систем управления базами данных на момент появления адаптивного подхода.

В настоящее время на рынке помимо реляционных подходов появилось множество решений, реализующих концепцию NoSQL, что позволило снять основные проблемы, из-за которых адаптивный подход был признан неэффективным. Это означает, что с использованием NoSQL-подхода можно построить решение для хранения данных, заточенное под определенный круг задач конкретной компании. Создание такого решения позволит как уменьшить стоимость разработки ПО, так и улучшить качество итогового продукта.

Рассмотрим минусы системы хранения адаптивной модели данных и ее отличия от баз данных NoSQL, чтобы показать, что с появлением NoSQL разрешились проблемы, мешавшие развитию адаптивной модели в реляционных базах.

Адаптивная модель хранения данных (AOM, Adaptive Object Model)

Написано множество статей, описывающих идею адаптивной модели хранения данных и ее модификации [2]. У сторонников AOM даже есть свой сайт, где собраны статьи, посвященные этому подходу. Можно увидеть, что все эти статьи написаны в 1990-е годы.

Заметим, что AOM создается под определенную предметную область и имеет свои особенности, позволяющие работать более эффективно именно в этой области. Мы рассмотрим базовую модель – ту часть, которая есть в любой модификации. Этого будет достаточно для понимания сути описываемого подхода.

Итак, начнем с краткого описания идеи адаптивного подхода к хранению данных. Концепция AOM вводит дополнительный уровень абстракции (помимо самих данных) – уровень метаданных. Метаданными называются данные, описывающие способ хранения данных; по сути, это информация о данных. Метаданные, как правило, составляют очень малую часть всего объема информации, но без них все данные превратятся в беспорядочную, неструктурированную кучу. В терминах AOM интересен именно метауровень, так как он отвечает за сложность системы с точки зрения как хранения, так и понимания, за гибкую подстройку системы к новым требованиям, за возможность конфигурирования.

Практически любые данные представляют собой набор объектов, их характеристик и описание их отношений между собой. В терминах AOM это Objects, Parameters, Relationships. Relationships можно рассматривать как особые параметры объекта, поэтому для упрощения модели будем говорить просто об объектах и их параметрах. Это некоторое абстрактное описание конкретных сущностей. Конкретика вводится с помощью метаинформации, и безликий объект становится объектом определенного типа, или ObjectType. Таким образом происходит обобщение всех возможных объектов неким обобщенным Object с присвоением ему определенного ObjectType.

Приведем пример того, как классические модели могут быть отражены на адаптивной модели. Под классическими мы будем понимать реляционные модели. Рассмотрим пример обычной таблицы в базе данных.

Итак, допустим, существует таблица Customers, в которой находятся несколько объектов: Customer1, Customer2 и т. д. Обобщение с помощью адаптивной модели даст две таблицы: одну с данными, другую с метаданными. В итоге в таблице Objects будет храниться Customer1, Customer2 с указанием того, что они имеют тип Customer; также в таблице ObjectTypes появится новая строка, говорящая о возможности существования объектов с таким типом. Любой другой объект также можно разместить в этих двух таблицах – Objects и ObjectTypes. То есть содержимое всех таблиц сливается в одну таблицу Objects, а описание – в таблицу ObjectTypes. Здесь ObjectTypes – это метаданные, Objects – данные.

Далее рассмотрим трансформацию параметров объекта на примере одной таблицы Customers. В данном случае данные, расположенные в столбцах, обобщаются с точки зрения реляционной теории. Так, если предположить, что в таблице Customers существуют столбцы City, Phone, то в данном случае описание параметров (возможность иметь City, Phone) является метаданными, или Attributes, а сами значения – данными, или Parameters. То есть Attributes будет иметь два значения: City, Phone. Для каждого Customer'а таблица Parameters может иметь значение для данного объекта и данного атрибута.

Очевидно, что данных в таблицах АОМ в разы больше, чем в оригинальных таблицах. Например, количество строк в таблице Objects равно суммарному числу строк всех изначальных таблиц, а в таблице ObjectTypes равно количеству изначальных таблиц. Количество строк в таблице Attributes равно суммарному числу столбцов во всех таблицах. Количество строк в таблице Parameters равно числу непустых ячеек во всех таблицах (то есть примерно произведению количества строк в Attributes на количество строк в Objects).

Причины возникновения баз данных NoSQL

Среди множества причин, послуживших толчком к возникновению баз данных NoSQL, можно выделить несколько наиболее значимых.

Первый момент – это скорее философский вопрос и предмет для дискуссий, нежели объективный критерий: несоответствие реляционной модели объектно-ориентированному уровню приложения (object-relational impedance mismatch) [3]. Сюда же можно отнести ориентированность стандарта на одного из производителей. Но все это не помешало SQL-решениям занять большую часть рынка. Однако исследования показывают, что технологии NoSQL постепенно распространяются все более широко [4].

Второй момент более объективный. В современном мире появилось много новых требований, связанных с изменением модели данных «на лету», которые ранее не возникали. Отсюда вытекает невозможность (или дороговизна) решений с zero-downtime.

Третий момент касается вопроса масштабирования. Большая часть реляционных баз была рассчитана на вертикальное масштабирование. О горизонтальном масштабировании особо не задумывались, так как только в последние годы объем информации начал расти лавинообразно.

Четвертый момент связан с серьезными изменениями в физической архитектуре: возросшие нагрузки подтолкнули к созданию кластеров из множества машин, поскольку необходимая совокупная производительность уже не могла быть реализована в одной физической машине. Поскольку реляционные базы были спроектированы и заточены под однопроцессорные машины, на первое место вышел вопрос технологий ПО. Например, к 2008 году инфраструктура Google состояла из 200 тысяч серверов [5], а реляционные хранилища изначально не создавались для такой инфраструктуры. Конечно, сейчас некоторые производители реляционных баз позиционируют себя как NoSQL-базы, поддерживающие горизонтальное масштабирование, но такие решения стоят очень дорого.

Пятый момент касается архитектуры приложения, которая изначально задумывалась как клиент-серверная. Клиент был «простой», и все задачи решались преимущественно на стороне сервера. В настоящее же время для эффективного решения возникающих задач необходим «умный» клиент – принимающий участие в решении задачи, делающий выводы из ответов сервера, создающий запросы на основе алгоритмов и логики.

Шестой момент, как уже частично упоминалось, связан с масштабом решаемых задач. Появились задачи, объем исходных данных для которых оказался на порядок больше, то есть стала необходима процессорная мощность сотен машин. Вертикально масштабируемые системы неспособны решить такие задачи. Также стоит упомянуть, что SQL-системы создавались для обработки транзакций, но это необходимо далеко не всегда: зачастую доступность гораздо важнее согласованности данных.

Седьмой момент – финансовый. Отчисления за лицензию на процессор или машину были слишком дороги, так как порой для обработки небольшого запроса требовались сотни узлов, что сразу сказывалось на стоимости инфраструктуры.

Применимость адаптивной модели для NoSQL-хранилищ

Теперь, когда определены особенности хранения и обработки данных в адаптивной модели и описана сама модель, можно сделать выводы о применимости адаптивных моделей, когда они используются вместе с базой данных NoSQL.

С точки зрения объектно-ориентированного подхода к практической реализации работы с адаптивными моделями необходимо ввести новый уровень абстракции, который поможет отойти от понятий «объект» и «атрибут» и позволит иметь дело со всем понятной объектной моделью. Как это должно быть реализовано, было подробно описано[6].

Проблема изменения модели на лету тоже становится неактуальной, так как набор и структура таблиц адаптивной модели не меняются. Добавление новой сущности или нового атрибута – это, по сути, добавление новой строки в таблицы метаданных: новая сущность – это новая строка в таблице объектных типов, новая характеристика сущности – это новая строка в таблице атрибутов.

Причина, по которой адаптивные модели плохо работают на реляционных системах, –

это отсутствие масштабируемости, так как при переходе на адаптивную модель объем данных растет в несколько раз быстрее, чем при разбиении данных по отдельным таблицам. Именно это и является основный причиной отказа от АОМ в реляционных хранилищах. В нереляционных хранилищах эта проблема решена. Заметим, что зависимость объема строк в таблицах от количества хранимых сущностей по-прежнему линейная как для Objects, так и для Parameters. То есть если в реляционных хранилищах стараются разбить все данные на как можно большее количество частей, то в NoSQL такой задачи не стоит. Там это делается на уровне платформы и называется шардингом [7]. Единственный момент, на который надо обратить внимание, – это возможность поддержки шардинга таблицами адаптивной модели, что невозможно при наличии ограничений типа «внешний ключ».

Таким образом, можно сделать вывод о применимости адаптивных моделей в NoSQL-среде с учетом правил, которые должны соблюдаться при разработке такого рода продуктов. Такой подход оптимален для систем, разрабатываемых на заказ и имеющих значительную общую часть функционала. Описанный подход не подходит для стартапов, где нет возможности вкладывать ресурсы в разработку платформы.

Список литературы

1. Your Ultimate Guide to the Non-Relational Universe! List Of NoSQL Databases. Available at: http://nosql-database.org/
2. Adaptive Object Model. Available at: http://adaptiveobjectmodel.com/writings/
3. Ted Neward's Technical Blog (2006–06–26). The Vietnam of Computer Science. Available at: http://blogs.tedneward.com/2006/06/26/The+Vietnam+Of+Computer+Science.aspx
4. DB-engines (Knowledge Base of Relational and NoSQL Database Management Systems). RDBMS dominate the database market, but NoSQL systems are catching up. Available at: http://db-engines.com/en/blog_post/23
5. CNET. Google spotlights data center inner workings. Available at: http://www.cnet.com/news/google-spotlights-data-center-inner-workings/
6. Васильев А.В. Шаблон проектирования корпоративных Java-приложений, построенных на основе адаптивных моделей данных, обеспечивающий их масштабируемость // Труды МФТИ. – 2013. – Т. 5, № 4 (20). – С. 96–101.
7. Sadalage P.J., Fowler M. 4: Distribution Models. NoSQL Distilled. 2012.

МОДЕЛИРОВАНИЕ И ДЕТЕКТИРОВАНИЕ ПРОХОЖДЕНИЯ РАДИОСИГНАЛА В ПОМЕЩЕНИИ

[1]Петров М.Н., [1]Акишев М.С., [2]Сафронов В.Д.

[1]Сибирский государственный аэрокосмический университет, Красноярск, e-mail: mnp_kafaes@mail.ru;
[2]Санкт-Петербургский государственный университет телекоммуникаций, Санкт-Петербург

В данной работе приведены экспериментальные результаты, на канальном уровне, распростронения и детектирование сигнала в помещении в условиях с замиранием Релея. Применение частотных эквалайзеров: с принудительным обнулением коэффициентов и с минимальной среднеквадратической ошибкой.

Распространяемый электромагнитный сигнал в помещении и вне его, подвержен отражениям, рассеяниям и дифракции, которые прямо пропорционально влияют на зависимость отношения значения сигнал шум к вероятности возникновения ошибки (BER) в беспроводном канале связи (например, сотой связи).

Такой сигнал на приёме получают, как несколько отражённых с разным временем задержки и мощностью сигналов, которые суммируясь, гасят друг друга. Такая вариация сигнала или многолучевой эффект в канале связи в помещении, где нет прямой видимости(NLOS) между передатчиком и приёмником, называется Релеевским замиранием. В помещении скорость в канале связи низкая из за, почти неподвижности пользователей (приёмника). Иногда значение мощности сигнала при глубоком замирании падает в несколько тысяч раз или 30-40 дБ.

Для генерации моделей канала, для мобильных беспроводных приложений используется квазистатический Релеевский канал с плоским замиранием. Замирание называется плоским тогда, когда ширина полосы пропускания (BW) сигнала меньше когерентной BW, а период символа больше разброса задержки. В квазистатическом медленном замирании канал изменяется независимо от кадра к кадру, однако в течении длительности кадра канал не изменяется. Его используют в моделировании, для того чтобы определить средний показатель (поведение) канала среди реализаций многих других.

Офис

Рассмотрим систему мобильной связи с модуляцией QPSK в офисе бизнес центра Москва в г. Астана (рис. 1). Здание монолитное, фасадные стены из газоблочного кирпича обшитые керамикой, дающие на 1800-2700 МГц ослабление в 4-6дБ [4], стены внутренние гипсокартонные, каждая комната имеет окно, через которое сигнал попадает на приёмник. Хорошим приближением к такой среде будет канал с одним отводом (tap) [4] который имеет плоскую частотную характеристику, не меняющуюся по времени. Такую среду помещения (Indoor) описывает квазистатический канал Релея. Большую часть времени мобильные пользователи статичны, то есть, нет подвижности пользователей, что задает частоту Доплера (fd = 0). Средам внутри таких помещений присущи замирания и переотражёние сигналов, результирующие такие замирания, которые не могут быть преодолены на принимающей стороне. Подразумевается, что между передатчиком и приёмником нет прямой видимости (K=0), и моделирование будет проведено в Релеевском многолучевом канале с плоским замиранием.