Предисловие

Мы работаем в области обработки данных предприятий более двадцати лет и стали свидетелями многочисленных изменений в языках программирования, архитектурах, платформах и процессах. Тем не менее на протяжении всего этого времени одно оставалось неизменным — хранение данных в реляционных базах. Было несколько попыток изменить это положение, и некоторые из них достигли успеха в отдельных нишах рынка, но в целом хранение данных с точки зрения архитектуры всегда сводилось к использованию реляционных баз данных.

Стабильность такой ситуации обеспечивает много преимуществ. Данные организаций существуют намного дольше, чем программы, которые их обрабатывают (по крайней мере, так говорят многие люди, хотя нам доводилось видеть множество очень старых программ). Это неплохо, что существуют стабильные хранилища данных, понятные и доступные для многих платформ прикладного программирования.

Однако сейчас появился новый конкурент, носящий дерзкое имя NoSQL. Он появился в ответ на необходимость обрабатывать более крупные объемы данных, которая вынуждает нас сделать фундаментальный сдвиг в сторону построения более крупных аппаратных платформ, состоящих из кластеров обычных серверов. В связи с этим обострились старые проблемы, связанные с обеспечением эффективной работы прикладных программ в рамках реляционной модели данных.

Термин “NoSQL” выбран очень неудачно. Он относится ко многим новым нереляционным базам данных, таким как Cassandra, Mongo, Neo4J и Riak. Они содержат неструктурированные данные, работают на кластерах и обеспечивают компромисс между традиционной согласованностью и другими полезными свойствами. Сторонники баз данных NoSQL утверждают, что могут создавать намного более высококачественные, намного лучше масштабируемые и легче программируемые системы.

Можно ли это считать предзнаменованием конца эпохи реляционных баз данных или просто еще одним претендентом на трон? Наш ответ: “Ни тем ни другим”. Реляционные базы данных — это мощный инструмент, который будет использоваться еще много десятилетий, но в результате происходящих изменений эти базы будут не единственными. С нашей точки зрения, мы вступаем в эру многовариантной персистентности (Polyglot Persistence), в которой предприятия и даже отдельные приложения будут использовать для управления данными несколько технологий. В результате архитекторы баз данных будут вынуждены осваивать эти технологии и оценивать их в соответствии со своими потребностями. Если бы мы так не думали, то не стали бы тратить время и силы на эту книгу.

Книга содержит достаточно много информации, чтобы ответить на вопрос, ст*о*ят ли базы данных NoSQL серьезного анализа для использования в будущих проектах. Каждый проект имеет свои отличительные особенности, поэтому невозможно нарисовать простое дерево решений для выбора правильного способа хранения данных.

Вместо этого мы попытались изложить принципы работы баз данных NoSQL, чтобы вы могли сделать самостоятельные выводы, не прибегая к сканированию веб. Мы намеренно написали небольшую книгу, чтобы вы могли достаточно быстро освоить эту информацию. Разумеется, в книге нет решений на все случаи жизни, но она должна сузить область поиска, которую вы должны исследовать, и помочь вам правильно сформулировать свои вопросы.

Чем интересны базы данных NoSQL

Есть две причины, по которым люди рассматривают возможность использовать базы данных NoSQL.

- **Эффективность разработки приложений.** Большинство усилий, связанных с разработкой приложений, затрачиваются на отображение данных из структур, хранящихся в памяти, в реляционные базы данных. База данных NoSQL может обеспечить модель данных, лучше удовлетворяющую потребности приложения, упростив тем самым это взаимодействие и уменьшив количество кода, который необходимо написать, отладить и развить.
- **Крупномасштабные данные.** Организации ценят возможность хранить более крупные объемы данных и быстрее их обрабатывать. Они считают слишком затратным использовать для этого реляционные базы данных. Основная причина заключается в том, что реляционные базы данных предназначены для работы на одном компьютере, в то время как большие объемы данных и программы для их обработки экономнее хранить на кластерах, состоящих из многочисленных небольших и дешевых компьютеров. Многие базы данных NoSQL разработаны специально для кластеров, поэтому они лучше вписываются в сценарии обработки больших объемов данных.

Краткое содержание книги

Книга состоит из двух частей. В первой части излагаются основные концепции, которые, по нашему мнению, необходимо знать, чтобы правильно оценивать возможность использования баз данных NoSQL в своих проектах и понимать, чем они отличаются от остальных. Во второй части мы сосредоточились на реализации систем с базами данных NoSQL.

Глава 1 начинается с объяснения быстрого роста популярности баз данных NoSQL — необходимость обрабатывать более крупные объемы данных в больших системах стимулировала переход от вертикального масштабирования к горизонтальному масштабированию на кластерах. Этим объясняется важная особенность многих баз данных NoSQL — явное хранение емких структур тесно связанных между собой данных, доступных как одно целое. В нашей книге мы называем такие структуры *агрегатами* (aggregate).

В главе 2 описывается, как агрегаты проявляются в трех основных моделях данных NoSQL: базы данных типа "ключ–значение" и документные базы данных ("Модели "ключ–значение" и документные модели данных"), а также семейство столбцов ("Семейства столбцов"). Агрегаты обеспечивают естественное взаимодействие различных приложений. Это одновременно ускоряет работу на кластерах и облегчает программам доступ к данным. В главе 3 рассматривается недостаток агрегатов — сложность выражения отношений ("Отношения") между сущностями в разных агрегатах. Это естественным образом приводит к графовым базам данных ("Графовые базы данных"), модели данных NoSQL, не соответствующей принципам, ориентированным на агрегаты. Мы также рассматриваем общую характеристику баз данных NoSQL — отказ от использования схем ("Неструктурированные базы данных"), который обеспечивает повышенную гибкость, но не настолько высокую, как можно было бы предположить.

Описав аспекты моделей данных в базах NoSQL, мы перейдем к их распределению: в главе 4 описывается, как база распределяет данные по кластерам. Эта процедура распадается на фрагментацию ("Фрагментация") и репликацию, которая может выполняться по схеме "ведущий–ведомый" (master-slave) ("Репликация ведущий–ведомый") или быть одноранговой (peer-to-peer) ("Одноранговая репликация"). Определив модели распределения, мы можем перейти к изучению согласованности. Благодаря ориентации на кластеры базы данных NoSQL обеспечивают более широкий выбор вариантов согласованности по сравнению с реляционными базами данных. В главе 5 описывается, как найти компромисс между изменениями согласованности при обновлении ("Согласованность при обновлении") и чтении ("Согласованность при чтении"), ролью кворумов ("Кворумы") и даже долговечностью ("Ослабление требования долговечности"). Если вы уже слышали что-либо о базе данных NoSQL, то почти наверняка слышали о теореме CAP; ее смысл и применение описывается в разделе "Теорема CAP".

Перечисленные выше главы посвящены в основном принципам распределения и сохранения согласованности данных, а в следующих двух главах описывается набор важных инструментов, выполняющих эту работу. В главе 6 описываются метки версий, отслеживающие изменения и устраняющие несогласованность. В главе 7 приводится краткий очерк организации параллельных вычислений, ориентированных на кластеры, а значит, и на системы NoSQL.

Освоив эти концепции, мы перейдем к вопросам их реализации, рассматривая конкретные примеры баз данных, относящихся к четырем ключевым категориям: в главе 8 в качестве базы данных типа "ключ–значение" используется база Riak, в главе 9 в качестве примера документной базы рассматривается база MongoDB, в главе 10 в качестве примера базы данных "семейство столбцов" описывается база Cassandra, а в главе 11 изучается графовая база данных Neo4J. Следует подчеркнуть, что это не исчерпывающее описание — за его пределами осталось много тем. Наш выбор примеров не следует рассматривать как рекомендации. Нашей целью было дать вам почувствовать разнообразие вопросов, существующих в этой области, и продемонстрировать применение описанных выше концепций в разных технологиях баз данных. Вы увидите, какие программы вам придется написать для этих систем, и получите представление об основных идеях, лежащих в их основе.

Принято считать, что благодаря отсутствию схемы в базах данных NoSQL можно легко изменять структуры данных на протяжении всего срока функционирования соответствующего приложения. Мы с этим не согласны — неструктурированная база данных имеет неявную схему, которая требует соблюдения принципов ее изменения при реализации, поэтому в главе 12 объясняется перенос данных как в системах со строгими схемами, так и в неструктурированных системах.

Благодаря всему сказанному становится ясно, что база данных NoSQL — не отдельная сущность и не может заменить реляционную базу данных. В главе 13 описывается будущая эра многовариантной персистентности, в которой будут сосуществовать разные способы хранения данных, даже в одном и том же приложении. Глава 14 расширяет горизонты книги и описывает другие технологии, которые не рассмотрены в книге, но также могут быть частью мира многовариантной персистентности.

Владея всей этой информацией, вы сможете делать осознанный выбор технологий хранения данных, поэтому заключительная глава ("Выбор базы данных") содержит несколько советов, касающихся этого выбора. С нашей точки зрения, существуют два главных фактора — определение эффективной программной модели, в которой модель хранения данных хорошо согласована с приложением, и обеспечение эффективного и надежного доступа к данным. Поскольку эра баз данных NoSQL только начинается, мы еще не имеем хорошо определенных процедур и должны согласовывать возможности со своими потребностями.

Это лишь краткий обзор — мы совершенно сознательно ограничили объем книги, выбрав лишь самую важную информацию. Если вы решили глубоко изучить эти технологии, то вам следует продолжить обучение, но мы надеемся, что эта книга станет хорошей отправной точкой на вашем пути.

Мы также хотели бы подчеркнуть, что базы данных NoSQL образуют очень изменчивую область компьютерной индустрии. Изменения в ней происходят ежегодно — появляются новые функциональные возможности и базы данных. Мы сделали основной акцент на концепциях, потому что их понимание является важным независимо от изменений соответствующих технологий. Мы убеждены, что большинство из написанного нами будет иметь значение еще долгое время, но абсолютно уверены, что проверку временем пройдет не все.

Для кого предназначена книга

Целевой аудиторией книги являются люди, так или иначе использующие базы данных NoSQL. Ее можно использовать как для создания нового проекта, так и для совершенствования существующего.

Мы стремились дать вам достаточно информации, чтобы вы могли осознанно решить, нужна ли технология NoSQL для удовлетворения ваших потребностей и какие инструменты следует изучить глубже. В качестве основных читателей мы

представляли себе архитекторов или технических руководителей, но думаем, что эта книга будет полезной всем, кто занимается управлением программным обеспечением и хочет получить представление о новой технологии. Мы считаем, что книга может стать хорошей отправной точкой для разработчиков, желающих овладеть новой технологией.

Мы не углублялись в детали программирования и развертывания конкретных баз данных — оставим эти вопросы для специальных книг. Мы стремились экономить объем, чтобы книга получилась краткой. Такую книгу удобно читать в самолете: она не стремится ответить на все вопросы, а дает читателям представление о том, какие вопросы он должен поставить.

Если вы уже работаете в области баз данных NoSQL, то эта книга, вероятно, ничего не прибавит к вашим знаниям. Тем не менее она будет полезной, поскольку поможет вам объяснить другим основные принципы технологии NoSQL. Очень важно научиться объяснять основные принципы технологии NoSQL, особенно если вы пытаетесь уговорить кого-то рассмотреть возможность использования баз данных NoSQL в проекте.

Что такое базы данных

В этой книге мы следуем основным принципам классификации баз данных NoSQL по их моделям данных. Ниже приведена таблица, содержащая четыре модели данных и базы данных, соответствующие этим моделям. Это не исчерпывающий список — он лишь напоминает о базах данных, не рассмотренных в книге. Пока мы писали книгу, полный список баз данных NoSQL хранился на веб-сайтах `http://nosql-database.org` и `http://nosql.mypopescu.com/kb/nosql`. В каждой категории мы пометили курсивом базу данных, рассмотренную в соответствующей главе.

Мы стремились выбрать репрезентативный инструмент для каждой категории баз данных. Говоря о конкретных примерах, мы имеем в виду всю категорию, даже если рассматриваемая база данных является уникальной и не допускает обобщений. Мы рассмотрели по одному примеру из категорий баз данных типа “ключ–значение”, документных баз данных, семейств столбцов и графовых баз данных. Там, где это было возможно, мы упоминали другие продукты, которые могут удовлетворить конкретные потребности пользователей.

Эта классификация по моделям данных является полезной, но грубой. Границы между разными моделями данных, например, между базами данных типа “ключ–значение” и документыми (“Модели “ключ–значение” и документные модели данных”), часто размыты. Многие базы данных не укладываются в рамки одной категории; например, база данных OrientDB одновременно относится к категориям документных и графовых баз данных.

Пример базы данных	Модель данных
BerkeleyDB Key-Value (“Базы данных типа “ключ–значение”)	Ключ–значение (“Базы данных типа “ключ–значение”)
	LevelDB
	Memcached
	Project Voldemort
	Redis
	Riak
CouchDB Document (“Документные базы данных”)	MongoDB
	OrientDB
	RavenDB
	Terrastore
Amazon SimpleDB Column-Family (“Семейства столбцов”)	Cassandra
	HBase
	Hypertable
FlockDB Graph (“Графовые базы данных”)	HyperGraphDB
	Infinite Graph
	Neo4J
	OrientDB

Благодарности

В первую очередь мы благодарим наших коллег из компании ThoughtWorks, многие из которых применяли базы данных NoSQL в своих проектах на протяжении последних лет. Их опыт был основным стимулом как для написания нашей книги, так и источником информации о ценности этой технологии. Положительный опыт, полученный при работе с базами NoSQL, стал основой нашего убеждения в том, что эта технология является фундаментальным сдвигом в области хранения данных.

Мы хотели бы поблагодарить разные группы людей, участвовавших в публичных дискуссиях, а также публиковавших статьи и блоги, посвященные использованию баз данных NoSQL. Основной прогресс в области разработки программного обеспечения остается скрытым, если люди не делятся своим опытом с коллегами. Особую благодарность выражаем компаниям Google и Amazon, опубликовавшим статьи о базах данных Bigtable и Dynamo. Эти статьи оказали большое стимулирующее влияние на развитие технологии NoSQL. Мы благодарим компании, финансирующие разработку баз данных NoSQL с открытым исходным кодом. Интересным отличием от предыдущих изменений в хранении данных является то, что технология NoSQL разрабатывалась как проект с открытым исходным кодом.

Отдельное спасибо компании ThoughtWorks за то, что она предоставила нам время для работы над книгой. Мы поступили на работу в компанию ThoughtWorks примерно в одно и то же время и работали в ней более десяти лет. Компания ThoughtWorks продолжает быть для нам радушным приютом, источником знаний и практического

опыта, а также гостеприимной средой для открытого обмена знаниями. Этим она сильно отличается от традиционных организаций, поставляющих системы баз данных.

Бетани Андерс–Бек (Bethany Anders-Beck), Илияс Бартолини (Ilias Bartolini), Тим Берглунд (Tim Berglund), Дункан Крейг (Duncan Craig), Поль Дюваль (Paul Duvall), Орен Эйни (Oren Eini), Перрин Фаулер (Perryn Fowler), Майкл Хангер (Michael Hunger), Эрик Кашич (Eric Kascic), Джошуа Кериевски (Joshua Kerievsky), Ананд Кришнасвами (Anand Krishnaswamy), Бобби Нортон (Bobby Norton), Аде Ошини (Ade Oshineye), Тьягу Паланисвами (Thiyagu Palanisamy), Прасанна Пендсе (Prasanna Pendse), Дан Притчетт (Dan Pritchett), Дэвид Райс (David Rice), Майк Робертс (Mike Roberts), Марко Родригес (Marko Rodriquez), Эндрю Слокум (Andrew Slocum), Тоби Трипп (Toby Tripp), Стив Виноски (Steve Vinoski), Дин Вамплер (Dean Wampler), Джим Уэббер (Jim Webber) и Уи Виттаваскул (Wee Witthawaskul) прочитали первые черновики книги и помогли нам своими советами.

Кроме того, Прамод хотел бы поблагодарить библиотеку Шамбурга (Schaumburg Library) за прекрасное обслуживание и уютное место для работы, прекрасных дочерей Архану и Арулу за понимание того, что папа должен идти в библиотеку и не может их взять с собой, а также любимую жену Рупали за безмерную поддержку и помощь.