

Предисловие

Когда я впервые окунулся в область действия больших данных, то почувствовал себя, как на Диком Западе в отношении разработки программного обеспечения. Многие разработчики переходили от реляционных баз данных с их привычными удобствами к нереляционным базам данных типа NoSQL с их ограниченными моделями данных, предназначенными для распределения по тысячам машин. С тех пор число баз данных типа NoSQL заметно возросло, а их отличия стали едва различимы. Затем возник новый многообещающий проект под названием *Hadoop*, дававший возможность выполнять глубокий анализ огромных объемов данных. И не так-то просто было разобраться, как рационально пользоваться всеми этими новыми инструментальными средствами.

В то время я пытался разрешать затруднения, которые возникали в связи с масштабированием информационных систем в тех организациях, где мне приходилось работать. Архитектура этих систем была угрожающе сложна, она состояла из целой паутины совместно используемых реляционных баз данных, очередей, рабочих, главных и подчиненных компонентов. Информация в таких базах данных была подвержена искажению, и для противодействия этому явлению в приложениях существовал специальный код. И в этом отношении ведомые компоненты системы всегда отставали. Поэтому я решил изучить альтернативные технологии больших данных, чтобы найти более совершенное проектное решение для имевшейся архитектуры информационной системы.

Один печальный опыт на ранней стадии моей карьеры разработчика программного обеспечения глубоко сформировал мое представление о том, как следует строить архитектуру информационных систем. Один из моих сотрудников потратил несколько недель на сбор данных из Интернета в совместно используемой файловой системе. Он предполагал собрать достаточный объем данных, чтобы проанализировать их. Однажды, выполняя плановое сопровождение системы, я случайно стер все данные своего сотрудника, сведя на нет все его многонедельные труды.

Я, конечно, понимал, что совершил непоправимую ошибку, но как начинающий разработчик программного обеспечения еще не осознавал всех ее последствий. Следовало ли меня уволить за подобную халатность? Я отправил по электронной почте сообщение команде разработчиков, принеся в нем глубочайшие извинения за свою оплошность, и, к большому удивлению, все члены команды отнеслись ко мне с большим сочувствием. Я никогда не забуду, как мой сотрудник подошел к моему рабочему столу, похлопал меня по спине и сказал: “Поздравляю. Теперь ты стал профессиональным разработчиком программного обеспечения”.

В его шутовском заявлении скрывалась следующая прописная истина разработки программного обеспечения: мы не знаем, как создавать идеальное программное обеспечение. Ошибки вполне могут проникать в программное обеспечение, развертываемое для эксплуатации. Если приложение может делать запись в базе данных, то такая запись может быть сделана ошибочно. И когда я приступил к переделке архитектуры имевшейся информационной системы, то опирался на этот опыт. Мне было известно, что новая архитектура должна быть масштабируемой и устойчивой не только к аппаратным отказам, но и в какой-то степени к ошибкам, связанным с человеческим фактором.

В ходе переделки архитектуры упомянутой выше системы мне пришлось пойти по пути переосмысления всего, что мне казалось истинным в отношении баз данных и управления ими. В итоге я построил архитектуру, основанную на непостоянных данных и их пакетной обработке. И меня просто поразило, насколько простой оказалась новая система в сравнении с той, что была основана только на инкрементной (т.е. пошаговой) обработке данных. Все стало проще, включая операции, новые средства поддержки системы, восстановление после ошибок, связанных с человеческим фактором, и оптимизацию производительности. Такой подход оказался настолько общим, что его, казалось, можно было бы применить к любой информационной системе.

Но меня все же кое-что смущало. Проанализировав положение дел в остальной отрасли, я обнаружил, что почти никто не работает по аналогичным методикам. Напротив, архитектуры информационных систем отличались непомерной сложностью и наличием огромных кластеров баз данных, обновлявшихся в пошаговом режиме. А разработанный мною подход позволял устранить или значительно упростить многие сложности в подобных архитектурах.

В течение нескольких последующих лет я расширил свою методику, формализовав ее в то, что теперь называется *лямбда-архитектурой*. Когда я работал в начинающей компании VaskTure, наша команда из пяти разработчиков создала программный продукт для анализа социальных сетей, предоставлявший в реальном времени разнообразные наборы аналитических данных объемом свыше 100 Тбайт. Наша небольшая команда вела также развертывание, эксплуатацию и мониторинг на кластере системы, состоящей из сотней машин. Когда мы продемонстрировали свой продукт пользователям, их поразило, что он был создан всего пятью людьми. Они часто задавали вопрос: “Как столь малым числом людей можно было сделать так много?” Мой ответ был прост: “Главное — не то, что мы делаем, а то, что мы не делаем”. С помощью лямбда-архитектуры нам удалось избежать сложностей, которыми страдали традиционные архитектуры. Избежав этих сложностей, мы сумели значительно повысить производительность своего труда.

Движение, называемое *большими данными*, только усугубило сложности, десятилетиями существовавшие в архитектурах данных. Любая архитектура, основанная в основном на крупных постепенно обновляемых базах данных, будет страдать подобными сложностями, вызывая программные ошибки, тяжелые операции и снижение производительности. И хотя базы данных типа SQL и NoSQL нередко изображаются как совершенно противоположные или дублирующие друг друга, на самом основополагающем уровне они, по существу,

одинаковы. Они стимулируют применение одной и той же архитектуры с ее неизбежными сложностями. Сложность — зловерное существо, которое все равно укусит вас, признаете ли вы его или нет.

Эта книга была написана в результате моего желания поделиться своими знаниями лямбда-архитектуры и способами избежать сложностей, присущих традиционным архитектурам информационных систем. Мне бы хотелось иметь такую книгу под рукой, когда я только начинал работать с большими данными. Надеюсь, что вы, читатель, отнесетесь к этой книге как к приключению, в ходе которого вам предстоит пересмотреть то, что вам было до сих пор известно об информационных системах, а также обнаружить, что с большими массивами данных можно работать изящно, просто и занимательно.

Натан Марц

Благодарности

Эта книга не появилась бы без помощи и поддержки многих неравнодушных людей. Начну со своих родителей, которые привили мне с юных лет любовь к изучению и исследованию окружающего мира. Они всегда поощряли меня во всех моих карьерных начинаниях.

Мой брат Иорав также поощрял во мне с юных лет интерес к интеллектуальному труду. Я до сих пор помню, как он учил меня алгебре, когда я ходил в начальную школу. Он был одним из тех, кто впервые познакомил меня с программированием, обучая меня языку Visual Basic, который он преподавал в средней школе. Эти уроки зародили во мне страсть к программированию и предопределили мою карьеру.

Я безмерно благодарен Майклу Монтано (Michael Montano) и Кристоферу Голде (Christopher Golda) — основателям компании BackType. С того момента, как они взяли меня на работу в свою компанию, я получил немалую свободу в принятии решений. И эта свобода дала мне возможность в полной мере исследовать и использовать лямбда-архитектуру. Они никогда не ставили под сомнение ценность открытого кода, позволив мне сделать нашу технологию свободно доступной для всех желающих воспользоваться ею. Возможность разрабатывать программное обеспечение с открытым исходным кодом стала одной из главных привилегий в моей жизни.

Особой признательности заслуживают многие мои преподаватели из Стенфордского университета. В частности, Тим Рафгарден (Tim Roughgarden) — самый лучший из моих преподавателей — коренным образом улучшил мои способности строго анализировать, разбирать и решать трудные задачи. Пройти как можно больше курсов, которые он вел, было одним из самых удачных решений в моей жизни. Выражаю также признательность Монике Лам (Monica Lam) за то, что она привила мне способность оценить по достоинству язык Datalog. Многие годы спустя я сочетал язык Datalog с моделью распределенных вычислений MapReduce в своем первом значительном проекте Cascalog с открытым исходным кодом.

Крис Уэнсел (Chris Wensel) первым показал мне, что обработка данных в масштабе может быть изящной и производительной. Его библиотека Cascading изменила мое отношение к обработке больших данных.

Ни одна из моих работ не состоялась бы без первопроходцев в области больших данных. Особая благодарность в этой связи выражается Джеффри Дину (Jeffrey Dean) и Санджаю Гемавату (Sanjay Ghemawat) — за оригинальную статью по MapReduce; Джузеппе Декандиа (Giuseppe DeCandia), Денизу

Хасторану (Deniz Hastorun), Мадану Джампани (Madan Jampani), Гунавардхану Какулапати (Gunavardhan Kakulapati), Авинашу Лакшману (Avinash Lakshman), Алексу Пилчину (Alex Pilchin), Свамнатану Сивасубраманияну (Swaminathan Sivasubramanian), Питеру Восхоллу (Peter Vosshall) и Вернеру Фогельсу (Werner Vogels) — за оригинальную статью по Dynamo; а также Майклу Кафарелле (Michael Cafarella) и Дугу Каттингу (Doug Cutting) — за основание проекта Apache Hadoop.

Рич Хики (Rich Hickey) был одним из тех, кто больше всех вдохновлял мои первые начинания в программировании. Clojure стал одним из самых любимых моих языков программирования. Изучив его, я стал лучше программировать, оценив его практичность и простоту. Философский подход Рича к состоянию и сложности в программировании оказал на меня большое влияние.

Когда я приступал к работе над этой книгой, у меня практически не было для этого никакого опыта. Поэтому особой благодарности заслуживает Рени Грегуар (Renaë Gregoire), одна из ответственных редакторов в издательстве Manning Publications, за оказанную помощь в моих литературных начинаниях. Она прочно вбила мне в голову важность употребления примеров для разъяснения общих понятий и пролила немало света на эффективные приемы написания технической литературы. Навыки, которым она меня обучила, можно применять не только к написанию технической литературы, но и к ведению блогов, дискуссий и общению вообще. И я всегда буду благодарен ей за приобретение столь важного для меня жизненного опыта.

Качество этой книги оказалось бы несравнимо ниже без усилий моего соавтора Джеймса Уоррена. Он проделал титаническую работу по разъяснению теоретических положений и поиску лучших способов подачи материала. Своей ясностью эта книга обязана большим навыкам Джеймса в общении.

Мне было приятно работать с сотрудниками издательства Manning Publications. Они проявили немало терпения и пониманием, что на поиски правильного способа подачи материала на столь обширную тему требовалось время. В течение всей работы над книгой они оказывали мне всяческую помощь и поддержку, предоставляя необходимые ресурсы. Поэтому хотелось бы поблагодарить Марьяна Бейса (Marjan Bace) и Майкла Стивенса (Michael Stephens) и всех остальных сотрудников издательства за то, что они всячески помогали и направляли мою работу над книгой.

Я стараюсь извлечь для себя как можно больше пользы, изучая труды других авторов. Особое влияние оказали на меня книги Бредфорда Кросса (Bradford Cross), Клейтона Кристенсена (Clayton Christensen), Пола Грехэма (Paul Graham), Карла Сагана (Carl Sagan) и Дерека Сиверза (Derek Sivers).

И наконец, я не могу не выразить благодарность многим людям, которые просматривали, комментировали данную книгу и давали отзывы о ней по мере ее написания. Благодаря ответной реакции этих людей мы, авторы книги, неоднократно пересматривали, переписывали и перестраивали ее структуру до тех пор, пока не нашли эффективные способы подачи материала. Особая благодарность в этой связи выражается Аарону Колкорду (Aaron Colcord), Аарону Кроу (Aaron Crow), Алексу Холмсу (Alex Holmes), Аруну Джекобу (Arun Jacob), Азифу Джану (Asif Jan), Айону Синхе (Ayon Sinha), Биллу Грехэму (Bill Graham), Чарльзу Брофи (Charles Brophy), Дэвиду Бекуиту (David Beckwith), Деррику Бернсу (Derrick

Burns), Дугласу Дункану (Douglas Duncan), Хьюго Гарца (Hugo Garza), Джейсону Куркуксу (Jason Courcoux), Джонатану Эстерхази (Jonathan Esterhazy), Карлу Кунцу (Karl Kuntz), Кевину Мартину (Kevin Martin), Лео Половцу (Leo Polovets), Марку Фишеру (Mark Fisher), Массимо Иларио (Massimo Iario), Майклу Фогусу (Michael Fogus), Майклу Г. Ноллу (Michael G. Noll), Патрику Деннису (Patrick Dennis), Педро Феррере Бертрану (Pedro Ferrera Bertran), Филиппу Джанерту (Philipp Janert), Родриго Абреу (Rodrigo Abreu), Руди Бонефасу (Rudy Bonefas), Сэму Ритчи (Sam Ritchie), Сиве Калагарле (Siva Kalagarla), Сорену Макбету (Soren Macbeth), Тимоти Шкловски (Timothy Chklovski), Валиду Фариду (Walid Farid), а также Женьхуа Гуо (Zhenhua Guo).

Натан Марц

Я был поставлен в затруднительное положение, когда передо мной встала задача упомянуть с благодарностью всех, кто внес свой посильный вклад в работу над этой книгой. К сожалению, здесь нет места упомянуть всех этих людей. Тем не менее мне хотелось бы выразить особую благодарность следующим людям.

- Моей жене Вен-Йинг Фенг (Wen-Ying Feng) — за ее любовь и моральную поддержку не только в работе над этой книгой, но и во всем, что мы делаем вместе.
- Моим родителям Джеймсу и Гретте Уоррен — за бесконечную веру в меня и жертвы, на которые они готовы пойти ради меня при всякой возможности.
- Моей сестре Джулии Уоррен-Уланч (Julia Warren-Ulanch) — за блестящий пример, которым она мне служила и которому я мог следовать.
- Моим учительницам и наставницам Эллен Тоби (Ellen Toby) и Сью Геллер (Sue Geller) — за готовность всегда ответить на любой вопрос и продемонстрировать радость поделиться своими знаниями, а не только приобрести их.
- Чаку Ламу (Chuck Lam) — за то, что он обратился когда-то ко мне со следующим вопросом: “А ты когда-нибудь слышал о чем-то вроде Hadoop?”
- Моим друзьям и коллегам из компаний RockYou!, Storm8 и Bina, — за общий приобретенный опыт и возможность воплотить теорию на практике.
- Марьяну Бейсу, Майклу Стивенсу, Дженифер Стаут (Jennifer Stout), Рени Грегуар и всем сотрудникам издательства Manning Publications — за наставничество и терпение, проявленное в отношении сроков завершения этой книги.
- Рецензентам и первым читателям этой книги — за комментарии и критические замечания в отношении ясности подачи материала, что в конечном итоге послужило заметному улучшению качества книги.

И наконец, мне хотелось бы выразить особую признательность моему соавтору, Натану Марцу, за то, что он предложил принять участие в данном проекте. Я и до этого был в восхищении от его трудов, и поэтому совместная работа с ним только умножила мое уважение к его идеям, принципам и подходам. Я почел за честь быть его соавтором.

Джеймс Уоррен

Об этой книге

Для таких веб-служб, как социальные сети, веб-аналитика и интеллектуальная электронная коммерция, нередко требуется управление данными в таких масштабах, которые слишком велики для традиционной базы данных. Сложность этой задачи возрастает по мере увеличения масштабов и требований, и поэтому большие данные — это не просто дубликат реляционных баз данных или развертывание новомодной технологии. Правда, масштабируемость и простота не исключают друг друга, а лишь требуют разных подходов. Для хранения и обработки данных в системах больших данных используется много работающих параллельно машин, и это представляет главные трудности, незнакомые большинству разработчиков.

В этой книге поясняется, как строить подобные системы на основе архитектуры, в которой выгодно используется кластерное оборудование наряду с инструментальными средствами, специально предназначенными для фиксации и анализа данных в масштабе веб. В ней описывается масштабируемый и легко понятный подход к системам больших данных, которые может построить и сопроводить небольшая команда специалистов. Теория построения систем больших данных подается в этой книге на реальных примерах их реализации на практике.

Большие данные не требуют предварительного раскрытия для анализа крупномасштабных данных или инструментальных средств типа NoSQL. Знакомство с традиционными базами данных полезно, хотя и не обязательно. Цель данной книги — помочь читателю научиться анализировать информационные системы, разделяя сложные задачи на простые решения. Мы начнем изложение материала данной книги с рассмотрения основных принципов, а затем перейдем к необходимым свойствам каждого компонента архитектуры.

Краткое содержание книги

Рассмотрим вкратце содержание 18 глав данной книги. В главе 1 представлены принципы построения информационных систем и вкратце описывается лямбда-архитектура: обобщенный подход к построению любых информационных систем. В главах 2–17 обсуждаются все составляющие лямбда-архитектуры с попеременным изложением теории и демонстрацией примеров ее применения на практике. В главах, посвященных теории, демонстрируются понятия, которые остаются в силе независимо от существующих инструментальных средств, тогда как в главах, посвященных демонстрации примеров, применение этих понятий на практике иллюстрируется с использованием реальных инструментальных

средств. Но пусть вас не смущают названия глав — все они опираются на конкретные примеры.

В главах 2–9 основное внимание уделяется *уровню пакетной обработки* лямбда-архитектуры. Из этих глав вы узнаете о моделировании главного массива данных, употреблении пакетной обработки для создания произвольных представлений данных, а также о компромиссах между инкрементной и пакетной обработкой данных.

Главы 10–11 посвящены *уровню обслуживания*, предоставляющему доступ с малой задержкой к представлениям, получаемым на уровне пакетной обработки. Из этих глав вы узнаете о специализированных базах данных, в которые информация записывается лишь в массовом порядке, и при этом обнаружите, что такие базы данных намного проще, чем традиционные, а следовательно, они обладают превосходной производительностью, просты и надежны в эксплуатации.

Главы 12–17 посвящены *уровню ускорения*, компенсирующему большую задержку, возникающую на уровне пакетной обработки, для получения актуальных результатов всех запросов. Из этих глав вы узнаете о базах данных типа NoSQL, потоковой обработке данных и преодолении сложностей инкрементной обработки данных.

В главе 18 вам предоставляется возможность пересмотреть лямбда-архитектуру, чтобы восполнить оставшиеся пробелы в своих знаниях, приобретенных в предыдущих главах. Из этой главы вы узнаете о инкрементной пакетной обработке данных, разных вариантах основной лямбда-архитектуры и способах извлечения наибольшей пользы из ресурсов, имеющихся в вашем распоряжении.

Загружаемый код и условные обозначения

Исходный код, загружаемый в дополнение к данной книге, можно найти по адресу <https://github.com/Big-Data-Manning>. Мы предоставили этот исходный код для выполнения примера службы веб-аналитики SuperWebAnalytics.com.

Большая часть этого исходного кода представлена на страницах книги в пронумерованных по порядку листингах. Некоторые листинги аннотированы для разъяснения отдельных фрагментов кода. А в других местах книги фрагменты кода употребляются по мере надобности. Исходный код выделяется **моноширинным шрифтом**, как это обычно принято в технической литературе. А **полужирным моноширинным шрифтом** в листингах и фрагментах кода выделяются те части исходного кода, которые поясняются в тексте книги.

Об иллюстрации на обложке книги

На обложке данной книги изображена фигура французского починщика керамики (Le Racommodeur de Fiance) начала XIX века. Его особое ремесло состояло в починке побитых или расколотых горшков, чашек, мисок и прочей керамической и глиняной утвари. Он ходил по деревням, городам и поселкам по всей Франции, предлагая свои ремесленные услуги.

Эта иллюстрация взята из четырехтомного издания Сильвена Марешаля (Sylvain Maréchal) сборника региональных обычаев в одежде, существовавших во Франции в начале XIX века. Каждая иллюстрация в этом сборнике нарисована вручную и художественно раскрашена в цвете. Большое разнообразие коллекции

одежды из сборника Марешаля ярко напоминает о том, насколько город и деревня были культурно разобщены почти 200 лет назад. Встречая людей на улицах городов и в деревнях, по их одежде нетрудно было понять, где они живут, каким ремеслом занимаются и каково их социальное положение.

С тех пор стиль, форма и региональные отличия в одежде претерпели существенные изменения, а ее богатство и разнообразие поблекло. Теперь жителей разных континентов трудно отличить по одежде, не говоря уже о горожанах и селянах. Возможно, люди променяли культурные различия на более разнообразную личную жизнь, особенно под влиянием идущего быстрыми темпами технического прогресса.

Ныне, когда трудно отличить одну книгу на компьютерную тематику от другой, издательство Manning Publications славится своей изобретательностью и инициативностью в издании технической литературы с обложками книг, наглядно иллюстрирующими богатство и разнообразие региональной жизни два столетия назад, возвращая к жизни с помощью иллюстраций одежды из сборника Марешаля.

От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо, либо просто посетить наш веб-сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг.

Наши электронные адреса:

E-mail: info@williamspublishing.com
WWW: <http://www.williamspublishing.com>

Наши почтовые адреса:

в России: 127055, г. Москва, ул. Лесная, д.43, стр. 1
в Украине: 03150, Киев, а/я 152

