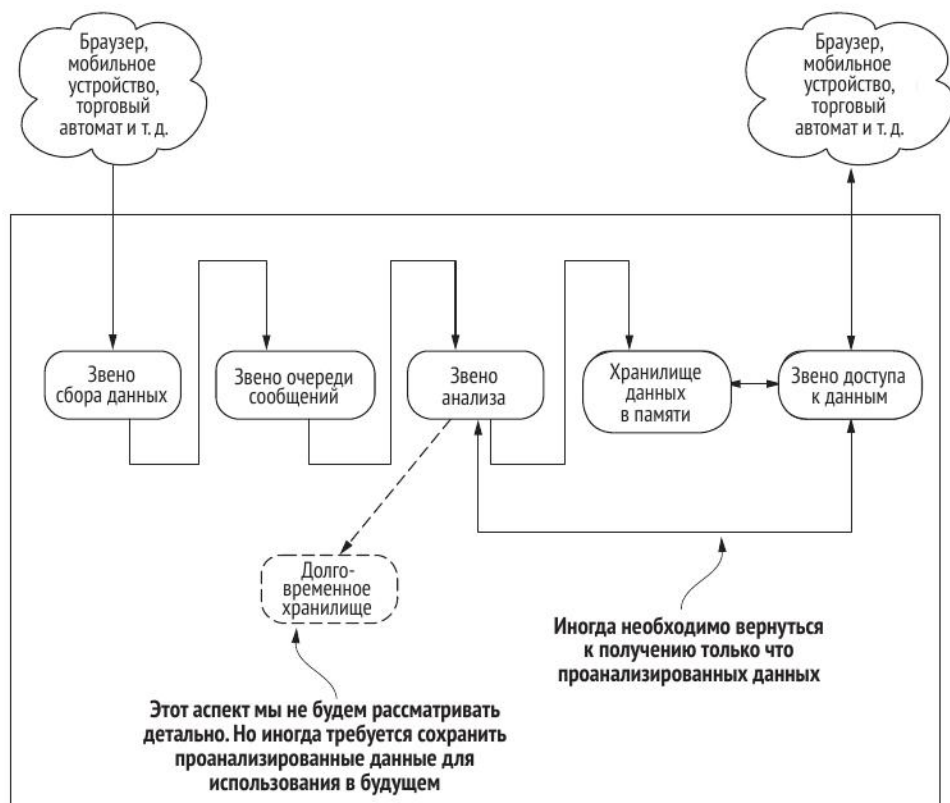


## Архитектурная диаграмма потоковой обработки данных



# Оглавление

Архитектурная диаграмма потоковой обработки данных .....	5
<b>Предисловие</b> .....	<b>9</b>
<b>Благодарности</b> .....	<b>11</b>
<b>Об этой книге</b> .....	<b>12</b>
Как работать с книгой?.....	12
Кому стоит прочитать эту книгу? .....	12
Структура книги.....	13
О коде .....	14
Об авторе .....	14
Автор в сети.....	15
Об иллюстрации на обложке.....	15
<b>Часть I. Новый целостный подход</b> .....	<b>17</b>
<b>Глава 1. Введение в потоковую обработку данных</b> .....	<b>19</b>
1.1. Что такое система реального времени?.....	20
1.2. Различия между системами реального времени и потоковыми системами.....	23
1.3. Архитектурная диаграмма .....	25
1.4. Безопасность в контексте потоковых систем.....	27
1.5. Как производится масштабирование? .....	27
1.6. Резюме .....	29
<b>Глава 2. Получение данных от клиентов: внесение данных</b> .....	<b>31</b>
2.1. Типичные паттерны взаимодействия .....	31
2.1.1. Запрос-ответ .....	32
2.1.2. Паттерн запрос-подтверждение .....	36
2.1.3. Паттерн издатель-подписчик.....	37
2.1.4. Паттерн одностороннего взаимодействия.....	39
2.1.5. Паттерн поток.....	40
2.2. Масштабирование паттернов взаимодействия .....	42
2.2.1. Паттерны запрос-ответ.....	43
2.2.2. Масштабирование паттерна поток .....	44
2.3. Отказоустойчивость.....	46
2.3.1. Протоколирование сообщений на стороне получателя .....	48
2.3.2. Протоколирование сообщений на стороне отправителя .....	51
2.3.3. Гибридное протоколирование сообщений.....	52
2.4. Опустимся на грешную землю .....	54
2.5. Резюме .....	55
<b>Глава 3. Транспортировка данных из звена сбора данных: расчленение конвейера данных</b> .....	<b>56</b>

3.1. Зачем нужно звено очереди сообщений .....	56
3.2. Основные концепции .....	58
3.2.1. Производитель, брокер и потребитель .....	59
3.2.2. Изоляция производителей от потребителей .....	61
3.2.3. Долговечные сообщения .....	62
3.2.4. Семантика доставки сообщений .....	65
3.3. Безопасность .....	69
3.4. Отказоустойчивость .....	70
3.5. Применение базовых концепций в конкретных задачах .....	73
3.6. Резюме .....	75
<b>Глава 4. Анализ потоковых данных .....</b>	<b>77</b>
4.1. Анализ данных в движении .....	77
4.2. Архитектуры распределенной обработки потоков .....	82
4.3. Ключевые функции систем потоковой обработки .....	88
4.3.1. Семантика доставки сообщений .....	89
4.4. Резюме .....	96
<b>Глава 5. Алгоритмы анализа данных .....</b>	<b>97</b>
5.1. Ограничения и их ослабление .....	98
5.2. К вопросу о времени .....	99
5.2.1. Скользящее окно .....	101
5.2.2. Прыгающие окна .....	103
5.3. Методы обобщения .....	106
5.3.1. Случайная выборка .....	106
5.3.2. Подсчет уникальных элементов .....	108
5.3.3. Частота .....	111
5.3.4. Вопрос о вхождении .....	113
5.4. Резюме .....	115
<b>Глава 6. Сохранение результатов сбора или анализа данных .....</b>	<b>116</b>
6.1. Когда нужно долговременное хранилище .....	118
6.2. Хранение данных в памяти .....	120
6.2.1. Встраиваемые хранилища в памяти с оптимизацией для флеш-памяти .....	121
6.2.2. Система кэширования .....	123
6.2.3. Базы данных и решетки данных в памяти .....	127
6.3. Примеры и упражнения .....	130
6.3.1. Сеансовая персонализация .....	130
6.3.2. Энергетическая компания следующего поколения .....	134
6.4. Резюме .....	135
<b>Глава 7. Получение доступа к данным .....</b>	<b>136</b>
7.1. Паттерны взаимодействия .....	137
7.1.1. Паттерн Data Sync .....	137
7.1.2. Удаленный вызов метода и удаленный вызов процедуры .....	139
7.1.3. Простой обмен сообщениями .....	140
7.1.4. Издатель-подписчик .....	141

7.2. Протоколы отправки данных клиентам .....	142
7.2.1. Веб-уведомления .....	143
7.2.2. Длинный HTTP-опрос .....	144
7.2.3. События, посылаемые сервером .....	146
7.2.4. Веб-сокеты .....	150
7.3. Фильтрация потока .....	154
7.3.1. Где производится фильтрация.....	154
7.3.2. Статическая и динамическая фильтрации .....	155
7.4. Пример: построение потокового API для сайта Meetup .....	156
7.5. Резюме.....	158
<b>Глава 8. Возможности конечных устройств и ограничения</b>	
<b>доступа к данным .....</b>	<b>160</b>
8.1. Основные концепции .....	162
8.1.1. Достаточная скорость чтения.....	163
8.1.2. Запоминание состояния .....	166
8.1.3. Смягчение последствий потери данных.....	168
8.1.4. Обработка ровно один раз .....	170
8.2. Все по-настоящему: компания SuperMediaMarkets .....	172
8.3. Введение в веб-клиент.....	176
8.3.1. Интеграция со службой потокового API .....	178
8.4. На пути к языку запросов .....	180
8.5. Резюме .....	181
<b>Часть II. Потоки в реальном мире .....</b>	<b>182</b>
<b>Глава 9. Анализ приглашений Meetup.com в режиме реального</b>	
<b>времени .....</b>	<b>183</b>
9.1. Звено сбора данных .....	185
9.1.1. Диаграмма последовательности службы сбора данных .....	185
9.2. Звено очереди сообщений .....	195
9.2.1. Установка и настройка Kafka .....	195
9.2.2. Интеграция службы сбора данных с Kafka .....	196
9.3. Звено анализа .....	198
9.3.1. Установка Storm и подготовка Kafka .....	199
9.3.2. Построение топологии Storm для нахождения <i>n</i> самых популярных тем.....	200
9.3.3. Интеграция звена анализа в конвейер .....	207
9.4. Хранилище данных в памяти .....	207
9.5. Звено доступа к данным .....	208
9.5.1. На пути к производственному режиму.....	213
9.6. Резюме .....	213
<b>Предметный указатель .....</b>	<b>214</b>

# Предисловие

Сколько себя помню, я всегда приходил в восторг от скорости обработки данных компьютером и стремился найти способ, как решить задачу быстрее. В конце 1990-х годов я в основном занимался программированием на C++ и моим любимым ключевым словом было `__asm`, означающее, что «следующий далее блок написан на языке Ассемблера». Я понимал, что происходит на машинном уровне. В начале 2000-х я работал над программным обеспечением мобильных устройств, и тогда снова стал актуальным вопрос, как быстрее синхронизировать данные или ускорить работу устройств PalmPilot и Windows CE? В то время мы имели дело с гигантскими (по тогдашним меркам, конечно) медицинскими базами данных (объемом 25–50 МБ), для хранения которых в PalmPilot нужно было вставлять внешнюю карту памяти, и разрабатывали приложения, которые должны были обеспечить быстрый интерактивный поиск и просмотр этих данных.

По мере роста объемов данных в интересующих меня отраслях я оказался в точке, где большие наборы данных сталкиваются с необходимостью их быстрой обработки для понимания сути дела. Данные генерировались во все ускоряющемся темпе, а бизнес хотел все быстрее получать ответы на вопросы, связанные с этими данными. Как раз то, что мне было надо: большие данные и жажда скорости. Где-то в 2001 году я начал работать над приложениями для анализа рынка и электронной коммерции, когда данные непрерывно обновлялись, а нам нужно было извлекать из них смысл почти в режиме реального времени. В 2009 году я перешел в компанию Webtrends, где моя любовь к скорости и своевременной обработке данных достигла зрелости. Основным бизнесом Webtrends была аналитика, и наши идеи, касающиеся аналитики в реальном времени, как раз стали вызывать интерес у заказчиков. Первый проект, над которым я работал, заключался в том, чтобы выводить ключевые показатели на информационную панель с задержкой не более пяти минут относительно потока данных о событиях на сайте, где бы они не происходили. В то время это выходило за рамки обыденности.

В 2011 году я вошел в группу разработки новых продуктов. Нашей задачей было и дальше продвигать идею аналитики в реальном времени и попытаться взорвать нашу отрасль изнутри. Мы потратили уйму времени на исследования, создание прототипа и обдумывание следующего шага, и тут случился идеальный шторм. Мы уже поглядывали в сторону Apache Kafka, когда в сентябре 2011-го был открыт исходный код Apache Storm. Мы тут же набросились на него с бешеным энтузиазмом. Уже зимой мы продемонстрировали то, чем занимались, нескольким заинтересованным

клиентам. В то время мы не оглядывались назад и всецело сосредоточились на выполнении соглашения об уровне услуг (Service Level Agreement, SLA), звучавшем примерно так: «от щелчка мышью в любой точке света до информационной панели – три секунды!» Спустя много месяцев труда куда большей по размеру командой мы выполнили свое обещание, и нас признали технологией года в области цифровой аналитики (Digital Analytics New Technology of the Year – [www.digitalanalyticsassociation.org/awards2013](http://www.digitalanalyticsassociation.org/awards2013)). Я был поглощен архитектурным проектированием и разработкой всех аспектов этого решения – от сбора данных до первого варианта пользовательского интерфейса (который ласково называл «голым скелетом», поскольку опыта в области построения UI у меня не было).

Мы не оставили свое увлечение и начали знакомиться с технологией Spark Streaming, когда она еще не вышла за стены Berkley AMPLab. С той поры я занимаюсь созданием все новых потоковых систем, стремясь к тому, чтобы полезная информация извлекалась из данных со скоростью мысли. Я выступаю на международных конференциях по этой тематике и работаю с компаниями по всему миру, помогая им проектировать и разрабатывать системы, решающие проблемы потоковой обработки данных.

Даже сегодня не все хорошо понимают, из каких частей состоит потоковая система. Найти информацию об отдельных компонентах нетрудно, но четкое понимание того, как устроен весь конвейер и как связаны между собой его звенья, встречается редко.

Поэтому я с огромным удовольствием попытался поделиться в этой книге своим практическим опытом и знаниями. Моя цель – предложить прочный фундамент, на котором можно строить и исследовать полноценные потоковые системы.

# Благодарности

Прежде всего хочу поблагодарить свою семью за поддержку во время работы над книгой. На протяжении многих выходных и вечеров я только и говорил: «Извините, я не могу помочь в саду (или поиграть в лакросс, или пойти в гости)». Конечно, и детям нелегко было это слышать, и жена устала все время брать на себя мою работу. Но на всем тернистом пути поддержка со стороны семьи ни разу не пошатнулась, она была для меня постоянным источником бодрости и вдохновения. Мой долг перед женой и детьми огромен, для его выражения одного «спасибо» недостаточно.

Спасибо Карен, моему редактору-консультанту, за бесконечное терпение, понимание и готовность в любое время обговаривать со мной различные вопросы. Спасибо Робину, рецензенту со стороны издательства, который поверил в меня, пестовал идею этой книги и служил слушателем, благодаря которому поезд не сошел с рельс на ранних этапах пути. Спасибо Берту, который учил меня рассказывать историю, находить подходящий уровень глубины и педагогически правильно выстраивать техническую книгу. Спасибо техническому редактору Грегору, чьи глубокие и полезные замечания помогли сделать книгу такой, какой вы ее видите. И наконец, спасибо всему коллективу издательства Manning за фантастическую работу, благодаря которой мы все-таки дошли до конца.

Спасибо всем, кто купил и прочитал ранние варианты рукописи в рамках программы предварительного ознакомления, всем, кто оставлял сообщения на форуме автора, а также многочисленным рецензентам за их бесценные замечания, в частности: Эндрю Джибсону (Andrew Gibson), д-ру Тобиасу Бюргеру (Tobias Bürger), Джейку Маккрэри (Jake McCrary), Родриго Абреу (Rodrigo Abreu), Эндт Кеффаласу (Andy Keffalas), Джону Гатри (John Guthrie), Космасу Чатзимихалису (Kosmas Chatzimichalis), Джулиано Бертоти (Giuliano Bertoti), Карлосу Куротто (Carlos Curotto), Энди Киршу (Andy Kirsch), Дугласу Данкану (Douglas Duncan), Джеффу Смиты (Jeff Smith) и Серджио Фернандесу Гонсалесу (Sergio Fernández González), Яромиру Д. Б. Немецу (Jaromir D. B. Nemes), Хосе Самонте (Jose Samonte), Яну Ноннену (Jan Nonnen), Ромиту Сингхаи (Romit Singhai), Крису Аллену (Chris Allan), Джонатану Томсу (Jonathan Thoms), Стивену Дженкису (Steven Jenkins), Лее Джилберту (Lee Gilbert), Амандипу Хурана (Amandeep Khurana), Чарли Гейнсу (Charlie Gaines). Без вас эта книга не состоялась бы.

Еще многие принимали участие в создании книги тем или иным способом, но я не могу упомянуть всех поименно, потому что так благодарности никогда бы не закончились. Тем не менее большое спасибо всем, кто помог мне на этом пути!

# Об этой книге

Системы реального времени появились давно, но в течение многих лет режим реального времени и потоковой обработки оставался вотчиной аппаратных систем. В таких системах невыполнение SLA угрожает человеческой жизни. Но за последние десять лет появились и стали быстро распространяться системы почти реального времени. Примеры потоковой обработки встречаются повсюду: социальные сети, игры, умные города, интеллектуальные измерительные устройства, ваша новая стиральная машина – список можно продолжать долго. Подумайте вот о чем: если считать, что байт равен одному галлону воды, то сегодня среднего размера дом можно было бы заполнить за 10 секунд, а к 2020 году это займет всего 2 секунды. Чтобы извлечь смысл из такого потока данных, нужны потоковые системы.

У этой книги, посвященной идеям потоковой обработки данных в режиме реального времени, две цели. Первая – научить вас рассуждать о конвейере в целом, чтобы вы могли не только построить потоковую систему, но и понимали, на какие компромиссы приходится идти в каждом ее звене. Вторая – заложить фундамент, опираясь на который, вы сможете глубже исследовать каждое звено, если это потребуется в интересах дела или просто ради удовлетворения любопытства.

## Как работать с книгой?

Предполагалось, что книгу будут читать с начала до конца, но каждая глава содержит достаточно информации, чтобы ее можно было прочитать и понять отдельно от других. Поэтому, если вы захотите узнать о конкретном звене, можете перейти прямо к соответствующей главе, а затем воспользоваться полученной информацией в качестве основы для более глубокого изучения остальных глав.

## Кому стоит прочитать эту книгу?

Эта книга рассчитана на разработчиков и архитекторов, но написана так, чтобы ее легко могли понять технические руководители и люди, принимающие решения о развитии бизнеса, – никакого предварительного знакомства с системами реального времени или потоковой обработки данных не предполагается. Единственное техническое требование – умение читать код, написанный на Java. На этом языке написаны как сами рассматриваемые системы, так и пример кода в главе 9.

## Структура книги

Структура книги изображена на рис. 1.

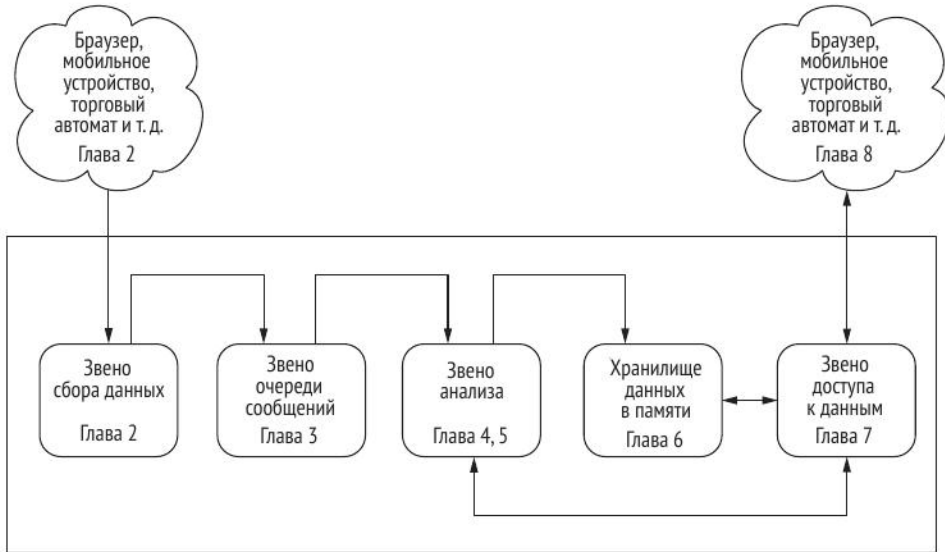


Рис. 1. Архитектурная диаграмма с разбиением на главы

*Глава 1* содержит краткий план всей книги, он может послужить в качестве карты на случай, если понадобится найти нужное звено. Вслед за планом определяется, что такое система реального времени, в чем состоит различие между системой реального времени и своевременной системой, и кратко затрагивается вопрос о важности безопасности (которому можно посвятить отдельную книгу).

*В главе 2* рассматриваются все аспекты сбора данных для потоковой системы: интерактивные средства, масштабирование и отказоустойчивость. Эта глава содержит все, относящееся к звену сбора данных. Прочитав ее, вы будете готовы к созданию надежного и масштабируемого звена.

*Глава 3* посвящена тому, как разорвать связь между собираемыми и анализируемыми данными, разместив между ними звено очереди сообщений. Вы узнаете, зачем нужно это звено, что такое долговечность сообщений, какие бывают семантики доставки сообщений и как выбрать технологию, подходящую для решения поставленной перед вами задачи.

*В главе 4* мы займемся распространенными архитектурными паттернами распределенной потоковой обработки и рассмотрим, что означает семантика доставки сообщений в этом звене, как работать с состоянием, что такое отказоустойчивость и почему она необходима.

*В главе 5* мы перейдем от обсуждения архитектуры к опросу потока, проблемам, касающимся времени, и четырем популярным техникам

обобщения. Если глава 4 отвечает на вопрос, *что* такое распределенная система потоковой обработки, то в главе 5 рассматривается ответ на вопрос, *как* она устроена.

В главе 6 обсуждаются варианты хранения данных в памяти в процессе анализа и после него. Мы не будем тратить времени на обсуждение методов долгосрочного хранения данных на диске, потому что такие решения зачастую используются за рамками потокового анализа и не способны обеспечить такую же производительность, как хранилища в памяти.

В главе 7 мы начнем разговор о том, что делать с данными, которые мы собрали и проанализировали. Здесь обсуждаются варианты коммуникации и протоколы отправки данных потоковому клиенту. Попутно мы ответим на вопрос, как бизнес-требования соотносятся с различными протоколами и как выбрать подходящий протокол.

В главе 8 речь идет о концепциях, которые следует иметь в виду при построении потокового клиента. Эта глава не просто о том, как разработать веб-приложение на основе HTML, в ней рассматриваются гораздо более глубокие, низкоуровневые вопросы проектирования клиентской части потоковой системы.

Глава 9... что ж, если вы прочитали все, что ей предшествует, принимайте поздравления! В первых восьми главах было рассмотрено много материала. А в главе 9 мы претворим теорию в практику – построим полный конвейер потоковой обработки данных и обсудим, как сделать из демонстрационного примера производственное приложение.

## О коде

Весь код, представленный в последней главе, можно найти в сети. Код можно скачать бесплатно с сайта издательства Manning по адресу [www.manning.com/books/streaming-data](http://www.manning.com/books/streaming-data) или найти его на GitHub по адресу <https://github.com/apsaltis/StreamingData-Book-Examples>.

Код разбит на отдельные Maven-проекты, по одному для каждого рассматриваемого в главе 9 звена. Инструкции по сборке и запуску программ приводятся по ходу дела.

Исходный код в листингах и внутри текста набран моноширинным шрифтом. Некоторые листинги аннотированы, чтобы пояснить ключевые моменты.

## Об авторе

Эндрю Пселтис одержим потоковыми системами, и все его устремления направлены на то, чтобы извлекать смысл из данных со скоростью мысли. Большую часть своего времени (когда не спит) он размышляет о потоковых системах, пишет о них или разрабатывает их. Он помогает клиентам всех мастей строить или исправлять сложные потоковые системы, расска-