

# Содержание

---

<b>Предисловие</b>	15
<b>Благодарности</b>	18
<b>Об этой книге</b>	21
Краткое содержание книги	21
Загружаемый код и условные обозначения	22
Об иллюстрации на обложке книги	22
От издательства	23
<b>Глава 1. Новая парадигма для больших данных</b>	25
1.1. Структура книги	26
1.2. Масштабирование в традиционных базах данных	27
1.2.1. Масштабирование с помощью очереди	28
1.2.2. Масштабирование путем фрагментации базы данных	28
1.2.3. Проявление вопросов отказоустойчивости	29
1.2.4. Проявление вопросов искажения данных	30
1.2.5. Что пошло не так	30
1.2.6. Чем могут помочь технологии организации больших данных	30
1.3. Базы данных типа NoSQL не являются панацеей	31
1.4. Основные принципы	31
1.5. Желательные свойства системы больших данных	32
1.5.1. Надежность и отказоустойчивость	33
1.5.2. Малые задержки чтения и записи данных	33
1.5.3. Масштабируемость	33
1.5.4. Обобщение	33
1.5.5. Расширяемость	34
1.5.6. Запросы с произвольным доступом	34
1.5.7. Минимальное сопровождение	34
1.5.8. Отлаживаемость	35
1.6. Недостатки полностью инкрементных архитектур	35
1.6.1. Эксплуатационная сложность	36

1.6.2. Крайняя сложность достижения окончательной согласованности	36
1.6.3. Неустойчивость к ошибкам, связанным с человеческим фактором	38
1.6.4. Полностью инкрементная архитектура в сравнении с лямбда-архитектурой	39
1.7. Лямбда-архитектура	40
1.7.1. Уровень пакетной обработки	42
1.7.2. Уровень обслуживания	43
1.7.3. Уровни пакетной обработки и обслуживания обеспечивают почти все свойства информационной системы	44
1.7.4. Уровень укорения	45
1.8. Современные тенденции в развитии технологий	48
1.8.1. Быстродействие ЦП не повышается	48
1.8.2. Эластичные облака	49
1.8.3. Эффектная экосистема с открытым кодом для больших данных	49
1.9. Пример приложения SuperWebAnalytics.com	50
Резюме	51
<b>ЧАСТЬ I. УРОВЕНЬ ПАКЕТНОЙ ОБРАБОТКИ</b>	<b>53</b>
<b>Глава 2. Модель данных для больших данных</b>	<b>55</b>
2.1. Свойства данных	57
2.1.1. Необработанность данных	59
2.1.2. Неизменяемость данных	63
2.1.3. Вечная истинность данных	65
2.2. Модель, основанная на фактах, для представления данных	66
2.2.1. Примеры фактов и их свойства	67
2.2.2. Преимущества модели, основанной на фактах	69
2.3. Граф-схемы	72
2.3.1. Элементы граф-схемы	73
2.3.2. Потребность в осуществимой схеме	74
2.4. Полная модель данных для приложения SuperWebAnalytics.com	75
Резюме	76
<b>Глава 3. Иллюстрация модели данных для больших данных</b>	<b>77</b>
3.1. Причины применения каркаса сериализации	77
3.2. Каркас сериализации Apache Thrift	78

3.2.1. Узлы	79
3.2.2. Ребра	79
3.2.3. Свойства	80
3.2.4. Связывание всего вместе в объекты данных	81
3.2.5. Развитие осуществимой схемы	81
3.3. Ограничения, присущие каркасам сериализации	83
Резюме	84
<b>Глава 4. Хранение данных на уровне пакетной обработки</b>	<b>85</b>
4.1. Требования к хранению главного массива данных	86
4.2. Выбор решения для хранения данных на уровне пакетной обработки	87
4.2.1. Применение хранилища пар “ключ–значение” для главного массива данных	88
4.2.2. Распределенные файловые системы	88
4.3. Принцип действия распределенных файловых систем	89
4.4. Сохранение главного массива данных в распределенной файловой системе	91
4.5. Вертикальное разделение	93
4.6. Низкоуровневый характер распределенных файловых систем	94
4.7. Хранение главного массива данных из приложения SuperWebAnalytics.com в распределенной файловой системе	96
Резюме	97
<b>Глава 5. Иллюстрация хранения данных на уровне пакетной обработки</b>	<b>99</b>
5.1. Применение распределенной файловой системы HDFS	100
5.1.1. Недостатки небольших файлов	101
5.1.2. Переход на более высокий уровень абстракции	102
5.2. Хранение данных на уровне пакетной обработки с помощью библиотеки Pail	103
5.2.1. Основные операции в Pail	104
5.2.2. Сериализация объектов в “ведрах”	105
5.2.3. Выполнение пакетных операций средствами Pail	106
5.2.4. Вертикальное разделение средствами Pail	107
5.2.5. Форматы и уплотнение “ведерных” файлов	108
5.2.6. Преимущества библиотеки Pail	109
5.3. Хранение главного массива данных для приложения SuperWebAnalytics.com	110
5.3.1. Структурированное “ведро” для хранения объектов Apache Thrift	112

5.3.2. Основное “ведро” для приложения SuperWebAnalytics.com	113
5.3.3. Расчлененное “ведро” для вертикального разделения массива данных	113
Резюме	116
<b>Глава 6. Уровень пакетной обработки</b>	<b>119</b>
6.1. Мотивирующие примеры	120
6.1.1. Количество просмотров страницы во времени	120
6.1.2. Заключение о гендерной принадлежности	121
6.1.3. Фактор влияния	121
6.2. Вычисления на уровне пакетной обработки	122
6.3. Сравнение алгоритмов повторных и инкрементных вычислений	124
6.3.1. Производительность	126
6.3.2. Устойчивость к отказам, обусловленным человеческим фактором	127
6.3.3. Универсальность алгоритмов	128
6.3.4. Выбор разновидности алгоритма	128
6.4. Масштабируемость на уровне пакетной обработки	129
6.5. MapReduce – парадигма распределенных вычислений для больших данных	130
6.5.1. Масштабируемость	131
6.5.2. Отказоустойчивость	134
6.5.3. Универсальность MapReduce	134
6.6. Низкоуровневый характер MapReduce	137
6.6.1. Многоэтапные вычисления неестественны	137
6.6.2. Соединения очень трудно реализуются вручную	137
6.6.3. Логическое исполнение тесно связано с физическим	139
6.7. Конвейерные схемы для рассмотрения пакетных вычислений на более высоком уровне	140
6.7.1. Принципы построения конвейерных схем	140
6.7.2. Выполнение конвейерных схем средствами MapReduce	145
6.7.3. Объединяющие агрегаторы	147
6.7.4. Примеры конвейерных схем	147
Резюме	149
<b>Глава 7. Иллюстрация уровня пакетной обработки</b>	<b>151</b>
7.1. Иллюстративный пример	152
7.2. Типичные ограничения, скрывающиеся в инструментальных средствах обработки данных	154
7.2.1. Специальные языки программирования	154
7.2.2. Неудачно составляемые абстракции	155

7.3. Введение в JCasalog	156
7.3.1. Модель данных JCasalog	156
7.3.2. Структура запросов в JCasalog	157
7.3.3. Запрашивание нескольких массивов данных	159
7.3.4. Группировка и агрегирование	162
7.3.5. Пошаговое выполнение примера запроса	163
7.3.6. Специальные предикатные операции	166
7.4. Композиция	171
7.4.1. Объединение подзапросов	171
7.4.2. Динамическое создание подзапросов	172
7.4.3. Предикатные макрокоманды	175
7.4.4. Динамическое создание предикатов макрокоманд	177
Резюме	179
<b>Глава 8. Пример построения уровня пакетной обработки: архитектура и алгоритмы</b>	<b>181</b>
8.1. Разработка уровня пакетной обработки приложения SuperWebAnalytics.com	182
8.1.1. Поддерживаемые запросы	182
8.1.2. Пакетные представления	183
8.2. Краткий обзор процесса пакетной обработки данных	186
8.3. Ввод данных	188
8.4. Нормализация URL	188
8.5. Нормализация идентификаторов пользователей	189
8.6. Удаление дубликатов событий просмотра страниц	194
8.7. Вычисление пакетных представлений	194
8.7.1. Количество просмотров страниц во времени	194
8.7.2. Подсчет индивидуальных посетителей страниц во времени	196
8.7.3. Анализ показателя отказов от просмотра	196
Резюме	198
<b>Глава 9. Пример реализации уровня пакетной обработки</b>	<b>199</b>
9.1. Отправная точка	200
9.2. Подготовка процесса пакетной обработки данных	200
9.3. Ввод новых данных	201
9.4. Нормализация URL	205
9.5. Нормализация идентификаторов пользователей	206
9.6. Исключение дубликатов событий просмотра страниц	212

<i>Содержание</i>	11
9.7. Вычисление пакетных представлений	212
9.7.1. Количество просмотров страниц во времени	212
9.7.2. Количество индивидуальных посетителей во времени	215
9.7.3. Анализа показателя отказов от просмотра	217
Резюме	219
<b>ЧАСТЬ II. УРОВЕНЬ ОБСЛУЖИВАНИЯ</b>	221
<b>Глава 10. Организация уровня обслуживания</b>	223
10.1. Количественные показатели производительности на уровне обслуживания	225
10.2. Уровень обслуживания как решение проблемы выбора между нормализацией или денормализацией данных	228
10.3. Требования к базе данных на уровне обслуживания	229
10.4. Проектирование уровня обслуживания для приложения SuperWebAnalytics.com	231
10.4.1. Количество просмотров страниц во времени	232
10.4.2. Количество индивидуальных посетителей страниц во времени	232
10.4.3. Анализ показателя отказов от просмотра страниц	233
10.5. Сравнение лямбда-архитектуры с полностью инкрементным решением	234
10.5.1. Полностью инкрементное решение задачи подсчета количества индивидуальных посетителей страниц во времени	234
10.5.2. Сравнение с решением на основе лямбда-архитектуры	241
Резюме	242
<b>Глава 11. Иллюстрация уровня обслуживания</b>	243
11.1. Основные положения о базе данных ElephantDB	244
11.1.1. Создание представления в базе данных ElephantDB	244
11.1.2. Обслуживание представления в базе данных ElephantDB	245
11.1.3. Применение базы данных ElephantDB	246
11.2. Построение уровня обслуживания для приложения SuperWebAnalytics.com	248
11.2.1. Просмотры страниц во времени	248
11.2.2. Индивидуальные посещения страниц во времени	250
11.2.3. Анализ показателя отказов от просмотра страниц	251
Резюме	252

<b>ЧАСТЬ III. УРОВЕНЬ УСКОРЕНИЯ</b>	253
<b>Глава 12. Представления в реальном времени</b>	255
12.1. Вычисление представлений в реальном времени	257
12.2. Хранение представлений в реальном времени	258
12.2.1. Достижимая точность	259
12.2.2. Сохранение состояния на уровне ускорения	260
12.3. Трудности инкрементных вычислений	261
12.3.1. Достоверность теоремы CAP	261
12.3.2. Сложная взаимосвязь теоремы CAP с алгоритмами инкрементных вычислений	263
12.4. Асинхронные обновления в сравнении с синхронными	265
12.5. Истечение срока действия представлений в реальном времени	267
Резюме	269
<b>Глава 13. Иллюстрация представлений в реальном времени</b>	271
13.1. Модель данных базы данных Cassandra	271
13.2. Применение базы данных Cassandra	273
13.2.1. Расширенные возможности базы данных Cassandra	275
Резюме	276
<b>Глава 14. Организация очередей и обработка потоков</b>	277
14.1. Организация очередей	278
14.1.1. Серверы однопользовательских очередей	278
14.1.2. Многопользовательские очереди	280
14.2. Обработка потоков	281
14.2.1. Очереди и рабочие процессы	282
14.2.2. Препятствия, скрытые в модели очередей и рабочих процессов	283
14.3. Высокоуровневая обработка потоков	284
14.3.1. Модель Storm	284
14.3.2. Гарантирование обработки сообщений	289
14.4. Уровень ускорения приложения SuperWebAnalytics.com	291
14.4.1. Структура топологии	293
Резюме	294

<b>Глава 15. Иллюстрация организации очередей и обработки потоков</b>	<b>297</b>
15.1. Составление топологий средствами Apache Storm	297
15.2. Кластеры Apache Storm и развертывание топологии	300
15.3. Гарантирование обработки сообщений	302
15.4. Реализация подсчета индивидуальных посещений страниц во времени на уровне ускорения приложения SuperWebAnalytics.com	304
Резюме	308
<b>Глава 16. Микропакетная обработка потоков</b>	<b>309</b>
16.1. Достижение семантики “только однажды”	310
16.1.1. Строго упорядоченная обработка	310
16.1.2. Микропакетная обработка потоков	311
16.1.3. Топологии микропакетной обработки потоков	312
16.2. Основные понятия микропакетной обработки потоков	315
16.3. Расширение конвейерных схем для микропакетной обработки потоков	316
16.4. Завершение построения уровня ускорения в приложении SuperWebAnalytics.com	318
16.4.1. Просмотры страниц во времени	318
16.4.2. Анализ показателя отказов от просмотра страниц	319
16.5. Другой подход к анализу показателя отказов от просмотра страниц	324
Резюме	325
<b>Глава 17. Иллюстрация микропакетной обработки потоков</b>	<b>327</b>
17.1. Применение Trident	328
17.2. Завершение построения уровня ускорения в приложении SuperWebAnalytics.com	331
17.2.1. Просмотры страниц во времени	331
17.2.2. Анализ показателя отказов от просмотра страниц	334
17.3. Полностью отказоустойчивая микропакетная обработка с сохранением состояния в памяти	340
Резюме	342



<b>Глава 18. Лямбда-архитектура в деталях</b>	343
18.1. Определение информационных систем	343
18.2. Уровни пакетной обработки и обслуживания	345
18.2.1. Инкрементная пакетная обработка	346
18.2.2. Измерение и оптимизация использования ресурсов на уровне пакетной обработки	353
18.3. Уровень ускорения	359
18.4. Уровень запросов	359
Резюме	361
<b>Предметный указатель</b>	363