

Посвящается нашим семьям и родителям

Содержание

Об авторах	15
Предисловие	16
От издательства	19
Глава 1. Введение	20
1.1. Что такое распределенная система баз данных?	21
1.2. История распределенных СУБД	22
1.3. Различные способы доставки данных	24
1.4. Обещания распределенных СУБД	26
1.4.1. Прозрачное управление распределенными и реплицированными данными	27
1.4.2. Обеспечение надежности с помощью распределенных транзакций	29
1.4.3. Повышенная производительность	30
1.4.4. Масштабируемость	32
1.5. Вопросы проектирования	33
1.5.1. Проектирование распределенной базы данных	33
1.5.2. Контроль распределенных данных	34
1.5.3. Распределенная обработка запросов	34
1.5.4. Распределенное управление конкурентностью	34
1.5.5. Надежность распределенной СУБД	35
1.5.6. Репликация	35
1.5.7. Параллельные СУБД	35
1.5.8. Интеграция баз данных	36
1.5.9. Альтернативные подходы к распределению	36
1.5.10. Обработка больших данных и NoSQL	36
1.6. Архитектуры распределенных СУБД	37
1.6.1. Архитектурные модели для распределенных СУБД	37
1.6.1.1. Автономность	37
1.6.1.2. Распределение	39
1.6.1.3. Гетерогенность	39
1.6.2. Клиент-серверные системы	40
1.6.3. Одноранговые системы	42
1.6.4. Системы управления мультибазами данных	45
1.6.5. Облачные вычисления	47
1.7. Библиографические замечания	52
Глава 2. Проектирование распределенных и параллельных баз данных	53
2.1. Фрагментация данных	56
2.1.1. Горизонтальная фрагментация	58

2.1.1.1. Требования к дополнительной информации.....	58
2.1.1.2. Главная горизонтальная фрагментация.....	61
2.1.1.3. Производная горизонтальная фрагментация.....	67
2.1.1.4. Проверка корректности.....	71
2.1.2. Вертикальная фрагментация.....	72
2.1.2.1. Требования к дополнительной информации.....	73
2.1.2.2. Алгоритм кластеризации.....	75
2.1.2.3. Алгоритм расщепления.....	80
2.1.2.4. Проверка корректности.....	83
2.1.3. Гибридная фрагментация.....	83
2.2. Размещение.....	84
2.2.1. Дополнительная информация.....	86
2.2.2. Модель размещения.....	87
2.2.2.1. Полная стоимость.....	87
2.2.2.2. Ограничения.....	89
2.2.3. Методы решения.....	90
2.3. Комбинированные подходы.....	90
2.3.1. Методы секционирования, безразличные к рабочей нагрузке.....	91
2.3.2. Методы секционирования, учитывающие рабочую нагрузку.....	92
2.4. Адаптивные подходы.....	96
2.4.1. Обнаружение изменений рабочей нагрузки.....	97
2.4.2. Обнаружение проблемных участков.....	98
2.4.3. Инкрементная реконфигурация.....	98
2.5. Каталог данных.....	101
2.6. Заключение.....	102
2.7. Библиографические замечания.....	103
Упражнения.....	105

Глава 3. Контроль распределенных данных..... 109

3.1. Управление представлениями.....	110
3.1.1. Представления в централизованных СУБД.....	110
3.1.2. Представления в распределенных СУБД.....	113
3.1.3. Обслуживание материализованных представлений.....	115
3.2. Контроль доступа.....	121
3.2.1. Избирательный контроль доступа.....	122
3.2.2. Мандатный контроль доступа.....	125
3.2.3. Распределенный контроль доступа.....	127
3.3. Контроль семантической целостности.....	129
3.3.1. Централизованный контроль семантической целостности.....	131
3.3.1.1. Спецификация ограничений целостности.....	131
3.3.1.2. Проверка целостности.....	133
3.3.2. Распределенный контроль семантической целостности.....	136
3.3.2.1. Определение распределенных ограничений целостности.....	136
3.3.2.2. Проверка распределенных ограничений целостности.....	139
3.3.2.3. Итоги обсуждения распределенного контроля целостности.....	142
3.4. Заключение.....	143
3.5. Библиографические замечания.....	143
Упражнения.....	145

Глава 4. Распределенная обработка запросов	148
4.1. Общий обзор	149
4.1.1. Задача обработки запроса	149
4.1.2. Оптимизация запроса.....	152
4.1.2.1. Пространство поиска	152
4.1.2.2. Модель стоимости	153
4.1.2.3. Стратегия поиска	154
4.1.3. Уровни обработки запросов	155
4.1.3.1. Декомпозиция запроса	156
4.1.3.2. Локализация данных	157
4.1.3.3. Распределенная оптимизация	158
4.1.3.4. Распределенное выполнение	159
4.2. Локализация данных	160
4.2.1. Редукция для главной горизонтальной фрагментации	160
4.2.1.1. Редукция с помощью выборки.....	161
4.2.2. Редукция с помощью соединения.....	162
4.2.3. Редукция для вертикальной фрагментации	163
4.2.4. Редукция для производной фрагментации.....	165
4.2.5. Редукция для гибридной фрагментации.....	166
4.3. Порядок соединений в распределенных запросах	168
4.3.1. Деревья соединений	169
4.3.2. Порядок соединений.....	170
4.3.3. Алгоритмы на основе полусоединений.....	172
4.3.4. Сравнение соединения и полусоединения	176
4.4. Распределенная модель стоимости	177
4.4.1. Функции стоимости	177
4.4.2. Статистика базы данных	179
4.5. Оптимизация распределенных запросов.....	181
4.5.1. Динамический подход.....	181
4.5.2. Статический подход.....	185
4.5.3. Гибридный подход	188
4.6. Адаптивная обработка запроса.....	193
4.6.1. Процесс адаптивной обработки запросов.....	194
4.6.1.1. Отслеживаемые параметры	194
4.6.1.2. Адаптивные реакции	195
4.6.2. Вихревой подход	196
4.7. Заключение	197
4.8. Библиографические замечания	198
Упражнения.....	200
Глава 5. Распределенная обработка транзакций	203
5.1. Основные понятия и терминология	205
5.2. Распределенное управление конкурентностью	208
5.2.1. Алгоритмы на основе блокировки.....	209
5.2.1.1. Централизованный алгоритм 2PL	210
5.2.1.2. Распределенный 2PL.....	213

5.2.1.3. Управление распределенными взаимоблокировками	214
5.2.2. Алгоритмы на основе временных меток	217
5.2.2.1. Базовый алгоритм упорядочения временных меток	218
5.2.2.2. Консервативный УВМ-алгоритм	221
5.2.3. Многоверсионное управление конкурентностью	223
5.2.4. Оптимистические алгоритмы	225
5.3. Распределенное управление конкурентностью с помощью изоляции моментальных снимков	227
5.4. Надежность распределенных СУБД	230
5.4.1. Протокол двухфазной фиксации	232
5.4.2. Варианты 2PC	237
5.4.2.1. Протокол 2PC с предполагаемой отменой	239
5.4.2.2. Протокол 2PC с предполагаемой фиксацией	240
5.4.3. Обработка отказов узлов	241
5.4.3.1. Протоколы завершения и восстановления для 2PC	241
5.4.3.2. Протокол трехфазной фиксации	247
5.4.4. Разделение сети	248
5.4.4.1. Централизованные протоколы	251
5.4.4.2. Протоколы на основе голосования	251
5.4.5. Протокол достижения консенсуса Paxos	252
5.4.6. Архитектурные соображения	255
5.5. Современные подходы к горизонтальному масштабированию управления транзакциями	257
5.5.1. Spanner	258
5.5.2. LeanXcale	259
5.6. Заключение	260
5.7. Библиографические замечания	263
Упражнения	266
Глава 6. Репликация данных	270
6.1. Согласованность реплицированных баз данных	272
6.1.1. Взаимная согласованность	272
6.1.2. Взаимная согласованность и согласованность транзакций	274
6.2. Стратегии управления обновлениями	276
6.2.1. Энергичное распространение обновлений	276
6.2.2. Ленивое распространение обновлений	277
6.2.3. Централизованные методы	278
6.2.4. Распределенные методы	278
6.3. Протоколы репликации	279
6.3.1. Энергичные централизованные протоколы	279
6.3.1.1. Единственный главный узел с ограниченной прозрачностью репликации	280
6.3.1.2. Единственный главный узел с полной прозрачностью репликации	282
6.3.1.3. Ведущая копия с полной прозрачностью репликации	285
6.3.2. Энергичные распределенные протоколы	286
6.3.3. Ленивые централизованные протоколы	287

6.3.3.1. Единственный главный узел с ограниченной прозрачностью репликации	287
6.3.3.2. Единственный главный или ведущий узел с полной прозрачностью репликации	289
6.3.4. Ленивые распределенные протоколы	292
6.4. Групповая коммуникация	294
6.5. Репликация и отказы	298
6.5.1. Отказы и ленивая репликация	298
6.5.2. Отказы и энергичная репликация	298
6.6. Заключение.....	302
6.7. Библиографические замечания.....	303
Упражнения.....	304

Глава 7. Интеграция баз данных – системы управления

мультибазами данных.....	307
7.1. Интеграция баз данных	308
7.1.1. Методология проектирования снизу вверх.....	309
7.1.2. Сопоставление схем	313
7.1.2.1. Гетерогенность схем.....	316
7.1.2.2. Подходы на основе лингвистического сопоставления	317
7.1.2.3. Сопоставление на основе ограничений.....	319
7.1.2.4. Сопоставление на основе обучения	321
7.1.2.5. Комбинированные подходы к сопоставлению.....	321
7.1.3. Интеграция схем.....	322
7.1.4. Отображение схем.....	324
7.1.4.1. Создание отображения	324
7.1.4.2. Обслуживание отображений.....	330
7.1.5. Очистка данных.....	332
7.2. Обработка мультибазовых запросов.....	333
7.2.1. Проблемы обработки мультибазовых запросов.....	334
7.2.2. Архитектура обработки мультибазового запроса	336
7.2.3. Переписывание запросов с помощью представлений	338
7.2.3.1. Терминология Datalog.....	338
7.2.3.2. Переписывание в случае ГКП	339
7.2.3.3. Переписывание в случае ЛКП.....	340
7.2.4. Оптимизация и выполнение запроса	343
7.2.4.1. Моделирование гетерогенной стоимости	343
7.2.4.2. Гетерогенная оптимизация запроса	350
7.2.5. Трансляция и выполнение запроса.....	355
7.3. Заключение.....	358
7.4. Библиографические замечания.....	360
Упражнения.....	363

Глава 8. Параллельные системы баз данных

8.1. Цели	375
8.2. Параллельные архитектуры	378

8.2.1. Общая архитектура	379
8.2.2. Архитектура с общей памятью.....	380
8.2.2.1. Равномерный доступ к памяти (UMA).....	380
8.2.2.2. Неравномерный доступ к памяти (NUMA).....	381
8.2.3. Архитектура с общим диском	383
8.2.4. Архитектура без разделения ресурсов	384
8.3. Размещение данных	385
8.4. Параллельная обработка запросов	388
8.4.1. Параллельные алгоритмы обработки данных	388
8.4.1.1. Параллельные алгоритмы сортировки.....	389
8.4.1.2. Параллельные алгоритмы соединения.....	390
8.4.2. Оптимизация параллельных запросов.....	396
8.4.2.1. Пространство поиска	396
8.4.2.2. Модель стоимости.....	399
8.4.2.3. Стратегия поиска	400
8.5. Балансировка запроса	400
8.5.1. Проблемы параллельного выполнения	401
8.5.2. Внутриоператорная балансировка нагрузки	403
8.5.3. Межоператорная балансировка нагрузки	405
8.5.4. Внутризапросная балансировка нагрузки	406
8.6. Отказоустойчивость.....	410
8.7. Кластеры баз данных	412
8.7.1. Архитектура кластера баз данных.....	412
8.7.2. Репликация	414
8.7.3. Балансировка нагрузки.....	414
8.7.4. Обработка запросов	415
8.8. Резюме	418
8.9. Библиографические замечания	419
Упражнения.....	421
Глава 9. Управление данными в одноранговых системах	425
9.1. Инфраструктура	428
9.1.1. Неструктурированные P2P-сети	429
9.1.2. Структурированные P2P-сети	432
9.1.3. Суперодноранговые P2P-сети	437
9.1.4. Сравнение P2P-сетей	438
9.2. Отображение схем в P2P-системах.....	439
9.2.1. Попарное отображение схем.....	439
9.2.2. Отображение на основе методов машинного обучения	440
9.2.3. Отображение на основе общего согласия	441
9.2.4. Отображение схем методами информационного поиска.....	442
9.3. Запросы в P2P-системах	442
9.3.1. Получение первых k результатов.....	442
9.3.1.1. Базовые методы	443
9.3.1.2. Запросы типа «первые k» в неструктурированных системах	450
9.3.1.3. Запросы типа «первые k» в DHT-системах.....	452
9.3.1.4. Запросы типа «первые k» в суперодноранговых системах	455

9.3.2. Запросы с соединением.....	455
9.3.3. Запросы по диапазону.....	457
9.4. Согласованность реплик.....	460
9.4.1. Базовая поддержка в DHT.....	460
9.4.2. Актуальность данных в DHT.....	462
9.4.3. Урегулирование реплик.....	464
9.4.3.1. OceanStore.....	464
9.4.3.2. P-Grid.....	466
9.4.3.3. APPA.....	466
9.5. Блокчейн.....	468
9.5.1. Определение блокчейна.....	469
9.5.2. Инфраструктура блокчейна.....	471
9.5.2.1. Создание транзакции.....	471
9.5.2.2. Группировка транзакций в блоки.....	471
9.5.2.3. Консенсусная проверка блока.....	473
9.5.3. Блокчейн 2.0.....	474
9.5.4. Проблемы.....	475
9.6. Заключение.....	477
9.7. Библиографические замечания.....	478
Упражнения.....	480
Глава 10. Обработка больших данных.....	482
10.1. Распределенные системы хранения.....	485
10.1.1. Google File System.....	486
10.1.2. Сочетание объектного и файлового хранения.....	488
10.2. Каркасы для обработки больших данных.....	489
10.2.1. Обработка данных в MapReduce.....	490
10.2.1.1. Архитектура MapReduce.....	492
10.2.1.2. Языки высокого уровня для MapReduce.....	494
10.2.1.3. Реализация операторов базы данных в MapReduce.....	495
10.2.2. Обработка данных с помощью Spark.....	500
10.3. Управление потоковыми данными.....	505
10.3.1. Потоковые модели, языки и операторы.....	507
10.3.1.1. Модели данных.....	507
10.3.1.2. Модели и языки потоковых запросов.....	509
10.3.1.3. Потоковые операторы и их реализация.....	509
10.3.2. Обработка запросов к потокам данных.....	511
10.3.2.1. Выполнение оконного запроса.....	512
10.3.2.2. Управление нагрузкой.....	513
10.3.2.3. Обработка не по порядку.....	514
10.3.2.4. Многозапросная оптимизация.....	515
10.3.2.5. Параллельная обработка потоков данных.....	516
10.3.3. Отказоустойчивость СПД.....	520
10.4. Платформы для анализа графов.....	521
10.4.1. Разбиение графа.....	525
10.4.2. MapReduce и анализ графов.....	530
10.4.3. Специализированные системы анализа графов.....	531

10.4.4. Ориентированная на вершины пошагово-синхронная модель	534
10.4.5. Ориентированная на вершины асинхронная модель	537
10.4.6. Ориентированная на вершины модель сбора–обработки–распространения.....	540
10.4.7. Ориентированная на разделы пошагово-синхронная модель	541
10.4.8. Ориентированная на разделы асинхронная модель	542
10.4.9. Ориентированная на разделы модель сбора–обработки–распространения.....	543
10.4.10. Ориентированная на ребра пошагово-синхронная модель.....	543
10.4.11. Ориентированная на ребра асинхронная модель.....	544
10.4.12. Ориентированная на ребра модель сбора–обработки–распространения.....	544
10.5. Озера данных	544
10.5.1. Озера данных и хранилища данных	545
10.5.2. Архитектура.....	546
10.5.3. Проблемы.....	548
10.6. Заключение.....	549
10.7. Библиографические замечания.....	550
Упражнения.....	553
Глава 11. NoSQL, NewSQL и полихранилища.....	557
11.1. Причины появления NoSQL	558
11.2. Хранилища ключей и значений.....	560
11.2.1. DynamoDB.....	560
11.2.2. Другие хранилища ключей и значений.....	563
11.3. Документные хранилища	563
11.3.1. MongoDB	564
11.3.2. Другие документные хранилища.....	567
11.4. Хранилища с широкими столбцами	568
11.4.1. Bigtable	568
11.4.2. Другие хранилища с широкими столбцами	570
11.5. Графовые СУБД.....	570
11.5.1. Neo4j.....	571
11.5.2. Другие графовые базы данных.....	575
11.6. Гибридные склады данных.....	575
11.6.1. Многомодельные NoSQL-системы.....	575
11.6.2. СУБД типа NewSQL.....	576
11.6.2.1. F1	577
11.6.2.2. LeanXcale.....	578
11.7. Полихранилища.....	580
11.7.1. Слабо связанные полихранилища.....	580
11.7.1.1. BigIntegrator	581
11.7.1.2. Forward	583
11.7.1.3. QoX	584
11.7.2. Сильно связанные полихранилища	585
11.7.2.1. Polybase	586
11.7.2.2. HadoopDB	588

11.7.2.3. Estocada	589
11.7.3. Гибридные системы	590
11.7.3.1. Spark SQL	590
11.7.3.2. CloudMdsQL	592
11.7.3.3. BigDAWG	594
11.7.4. Заключительные замечания	594
11.8. Заключение	595
11.9. Библиографические замечания	597
Упражнения	598

Глава 12. Управление веб-данными

12.1. Управление веб-графом	601
12.2. Поиск в вебе	603
12.2.1. Обход веба роботом	604
12.2.2. Индексирование	607
12.2.2.1. Структурный индекс	607
12.2.2.2. Текстовый индекс	607
12.2.3. Ранжирование и анализ ссылок	608
12.2.4. Поиск по ключевым словам	609
12.3. Запросы к вебу	610
12.3.1. Веб как слабо структурированные данные	611
12.3.2. Языки веб-запросов	616
12.4. Вопросно-ответные системы	620
12.5. Поиск и опрос скрытого веба	625
12.5.1. Обход скрытого веба	625
12.5.1.1. Запрос через поисковый интерфейс	625
12.5.1.2. Анализ страниц результатов	626
12.5.2. Метапоиск	627
12.5.2.1. Выделение резюме содержимого	627
12.5.2.2. Категоризация баз данных	628
12.6. Интеграция веб-данных	629
12.6.1. Веб-таблицы и фьюжн-таблицы	630
12.6.2. Семантический веб и проект Linked Open Data	630
12.6.2.1. XML	633
12.6.2.2. RDF	636
12.6.2.3. Навигация и опрос в проекте LOD	647
12.6.3. Вопросы качества данных при интеграции веб-данных	648
12.6.3.1. Очистка структурированных веб-данных	649
12.6.3.2. Слияние веб-данных	651
12.6.3.3. Качество источника веб-данных	652
12.7. Библиографические замечания	655
Упражнения	658

Предметный указатель

660

Об авторах

М. Мамер Ёсу – профессор Черитонской школы компьютерных наук в университете Ватерлоо в Канаде. Исследованиями в области распределенного управления данными он занимается уже тридцать лет. Состоит членом Королевского общества Канады, Американской ассоциации содействия развитию науки (AAAS), Ассоциации по вычислительной технике (ACM) и Института инженеров по электротехнике и электронике (IEEE). Является избранным членом Турецкой академии наук и членом общества «Сигма Кси». Является лауреатом премии за прижизненные достижения Канадского общества компьютерных наук за 2019 год, премии за проверенные временем достижения ACM SIGMOD за 2015 год, премии за вклад в науку ACM SIGMOD за 2008 год и премии выдающимся выпускникам технического колледжа университета штата Огайо за 2008 год. Также получил две премии за лучшую работу и один похвальный отзыв на публикацию. Состоит в редколлегиях многих журналов и книжных серий, наряду с Линь Лю является одним из главных редакторов «Энциклопедии по системам баз данных».

Патрик Вальдуриес – главный научный сотрудник французской компании Inria. Преподавал информатику в университете Пьера и Мари Кюри (UPMC) в Париже (2000–2002) и занимал должность исследователя в компании Microelectronics and Computer Technology Corp. в Остине, штат Техас (1985–1989). Начиная с 2019 года является научным консультантом стартапа LeanXcale.

В настоящее время возглавляет команду Zenith (включающую сотрудников компании Inria и университета Монпелье, LIRMM), которая занимается наукой о данных, в частности управлением данными в крупномасштабных распределенных и параллельных системах и управлением научными данными. Является заместителем редактора в нескольких журналах, в частности *VLDB Journal*, *Distributed and Parallel Databases* и *Internet and Databases*.

Занимал место в правлении таких крупных конференций, как SIGMOD и VLDB. Исполнял обязанности председателя конференций SIGMOD 2004, EDBT 2008 и VLDB 2009. Получил несколько наград за лучшую работу, в т. ч. на VLDB 2000. Лауреат премии по информатике от французского подразделения IBM за 1993 год и премии компании Inria, Французской академии наук и компании Dassault Systems за инновации в 2014 году. Является действительным членом ACM.

Предисловие

Первое издание этой книги вышло в 1991 году, когда технология была новой, а продуктов не так много. В предисловии к первому изданию мы цитировали Майкла Стоунбрейкера, который в 1988 году говорил, что в следующие 10 лет централизованные СУБД станут «антикварной редкостью», а большая часть организаций перейдет на распределенные СУБД. Это предсказание, конечно же, сбылось, в современном мире значительная доля систем либо распределенные, либо параллельные – обычно для их описания употребляется термин «горизонтально масштабируемые». Когда мы собирали материал для первого издания, курсы по базам данных для студентов и магистрантов были не так распространены, как сейчас, поэтому книга содержала пространные сведения о централизованных решениях, предварявшие обсуждение распределенных и параллельных систем. Но и на этом фронте произошли большие перемены, теперь трудно встретить магистранта, не имеющего хотя бы начальных знаний о технологии баз данных. Поэтому учебник по технологии распределенных и параллельных баз данных для магистрантов сейчас нужно позиционировать иначе. Такую цель мы и поставили себе в этом издании, сохранив, впрочем, многие новые темы, появившиеся в третьем издании. Перечислим основные изменения, внесенные в четвертое издание.

1. С годами побудительные мотивы этой технологии и среда ее развертывания претерпели изменения (веб, облако и т. д.). Поэтому вводная глава нуждалась в серьезном пересмотре. Мы переписали введение, отразив современный взгляд на технологию.
2. Мы добавили новую главу, посвященную обработке больших данных, в которую включили распределенные системы хранения, потоковую обработку данных, платформы MapReduce и Spark, анализ графов и озера данных. По мере распространения таких систем приобретает важность систематическое изложение этих вопросов.
3. Аналогично мы учли растущее влияние NoSQL-систем, посвятив им отдельную главу. В ней дается обзор четырех типов баз данных NoSQL (хранилища ключей и значений, документные хранилища, столбцовые базы данных и графовые СУБД), а также NewSQL-систем и полихранилищ.
4. Мы объединили главы об интеграции баз данных и обработке запросов к нескольким базам в одну главу.
5. Мы кардинально переработали изложение вопроса об обработке данных в вебе, сместив акцент с XML на технологию RDF, которая сейчас больше распространена. В эту главу мы включили обсуждение подходов к интеграции веб-данных, в т. ч. важный вопрос о качестве данных.
6. Мы также подвергли ревизии главу об одноранговой обработке данных и включили подробное объяснение технологии блокчейн.
7. Стремясь вычистить предыдущие главы, мы ужали главы, относящиеся к обработке запросов и управлению транзакциями, исключив принципиально централизованные методы и сосредоточив внимание на рас-

предельных и параллельных. Попутно мы включили некоторые темы, которые приобрели большое значение, в частности динамическую обработку запросов (вихревых операторов), а также алгоритм консенсуса Paxos и его применение в протоколах фиксации.

8. Мы исправили главу о параллельных СУБД, уточнив цели, в частности пояснили различие между горизонтальным и вертикальным масштабированием и обсудили параллельные архитектуры, в т. ч. UMA и NUMA. Также добавлен новый раздел о параллельных алгоритмах сортировки и вариантах параллельных алгоритмов соединения, в которых задействованы преимущества памяти большого объема и многоядерных процессоров, которые ныне распространены повсеместно.
9. Мы пересмотрели главу о проектировании распределенности, включив пространное обсуждение современных подходов, сочетающих фрагментацию и размещение. После реорганизации материала эта глава стала центральным источником информации по секционированию данных во всех обсуждениях распределенного и параллельного управления данными.
10. Хотя объектные технологии по-прежнему играют роль в информационных системах, их значимость в системах распределенного и параллельного управления данными снизилась. Поэтому в этом издании мы исключили главу об объектных базах данных.

Как и в случае предыдущих изданий, в редактировании книги нам помогали многие коллеги, которым мы выражаем благодарность (не придерживаясь при перечислении какого-то определенного порядка). Дэн Олтеану (Dan Olteanu) предложил в главе 3 изящное обсуждение двух оптимизаций, которые могут значительно уменьшить время обслуживания материализованных представлений. Фил Бернштейн (Phil Bernstein) предоставил предварительные материалы к новым статьям о многоверсионном управлении транзакциями, легшие в основу обновленного обсуждения этой темы в главе 5. Хузаима Дауджи (Khuzaima Daudjee) также помог в подготовке списка более современных публикаций по распределенной обработке транзакций, который мы включили в библиографические ссылки к той же главе. Рикардо Хименес-Перис (Ricardo Jimenez-Peris) предоставил материал по высокопроизводительным транзакционным системам, включенный туда же. Деннис Шаша (Dennis Shasha) отредактировал новый раздел по блокчейну в главе по одноранговым системам. Майкл Кэри (Michael Carey) прочитал главы по большим данным, NoSQL, NewSQL, полихранилищам и параллельным СУБД и дал весьма подробные замечания, благодаря которым эти главы стали значительно лучше. Студенты Тамера, Анли Пачачи (Anil Pacaci), Халед Аммар (Khaled Ammar) и аспирант Сяофей Чжан (Xiaofei Zhang) написали подробные рецензии на главу по большим данным, и части их текстов включены в эту главу. В главу по NoSQL, NewSQL и полихранилищам включены части публикаций Бояна Колева (Boyan Kolev) и студентки Патрика Карлины Бондиомбой (Carlyna Bondiombooy). Джим Веббер (Jim Webber) отредактировал раздел по Neo4j этой главы. Характеристика графовых аналитических систем, приведенная в этой главе, частично основана на магистерской диссертации Миньянь Хана, в которой он заодно предложил подход на основе GiraphUC, обсуждаемый там же. Части этой главы прочитали и оставили весьма полезные замечания

также Семих Салихоглу (Semih Salihoglu) и Лукаш Голаб (Lukasz Golab). Алон Халеви поделился комментариями по поводу обсуждения веб-таблиц в главе 12. Обсуждение качества данных в процессе интеграции веб-данных написали Ихаб Ильяс (Ihab Ilyas) и Су Чу (Xu Chu). Стратос Идреос (Stratos Idreos) пояснил, как можно использовать крекинг базы данных в качестве подхода к секционированию, его текст включен в главу 2. Ренан Соуза (Renan Souza) и Фабиан Штёттер (Fabian Stöter) просмотрели всю книгу целиком.

В третье издание книги было включено много новых тем, которые перекочевали в это издание, при написании этих глав большой вклад внесли многие наши коллеги. Мы хотим еще раз отметить их содействие, поскольку его влияние прослеживается и в новом издании. Рене Миллер (Renée Miller), Эрхард Рам (Erhard Rahm) и Алон Халеви (Alon Halevy) многое сделали для сведения воедино материалов по интеграции баз данных, которые затем были внимательно отрецензированы Авигдором Галом (Avigdor Gal). Матиас Ярке (Matthias Jarke), Сянь Ли (Xiang Li), Готфрид Фоссен (Gottfried Vossen), Эрхард Рам и Андреас Тор (Andreas Thor) предложили упражнения к этой главе. Хуберт Наак (Hubert Naacke) внес вклад в раздел о моделировании гетерогенной стоимости, а Фабио Порто (Fabio Porto) – в раздел об адаптивной обработке запросов. Главу 6 о репликации данных невозможно было бы написать без помощи Густаво Алонсо (Gustavo Alonso) и Беттины Кемме (Bettina Kemme). Эстер Пачитти (Esther Pacitti) также внесла вклад в главу о репликации данных – прочитав ее и предложив вводные материалы; она же помогала в написании раздела о кластерах баз данных в главе о параллельных СУБД. При работе над главой об одноранговой обработке данных мы много беседовали с Бенг Чин Ои (Beng Chin Ooi). В разделе этой главы, посвященном обработке запросов в одноранговых системах, использованы материалы из докторской диссертации Резы Акбаринья (Reza Akbarinia) и Венцеслао Пальма (Wenceslao Palma), а в разделе о репликации – материалы из докторской диссертации Видала Мартинса (Vidal Martins).

Мы благодарны нашему редактору в издательстве Springer Сюзан Лагерстром-Файф (Susan Lagerstrom-Fife) за продвижение проекта в самом издательстве и за постоянные напоминания о необходимости закончить работу вовремя. Мы пропустили почти все назначенные ей крайние сроки, но надеемся, что результат получился неплохим.

Наконец, нам было бы очень интересно услышать ваши замечания и предложения. Мы приветствуем любые отзывы, но особенно нас интересует ваше мнение по следующим вопросам:

- 1) любые ошибки и опечатки, просочившиеся, несмотря на все наши усилия (хочется надеяться, что их немного);
- 2) темы, которые следовало бы исключить, и, наоборот, темы, которые нужно добавить или изложить более подробно;
- 3) придуманные вами упражнения, которые вы хотели бы включить в книгу.

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Springer очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Глава 1

Введение

Современная вычислительная среда в значительной степени распределенная – компьютеры подключены к интернету и образуют всемирную распределенную систему. В организациях имеются территориально распределенные связанные между собой центры обработки данных (ЦОД), насчитывающие сотни, а то и тысячи компьютеров, объединенных в высокоскоростную сеть, которая содержит как распределенные, так и параллельные системы (рис. 1.1). Объем данных, хранимых в такой среде, возрос многократно. Не все данные хранятся в системах баз данных (на самом деле там хранится лишь малая их часть), но возникает желание так или иначе управлять этим распределенным по большой территории массивом данных. Это и есть задача распределенных и параллельных систем баз данных, которые несколько десятилетий назад занимали лишь небольшую нишу в мировой вычислительной среде, а теперь вышли на авансцену. В данной главе мы дадим общий обзор этой технологии, а в последующих займемся деталями.

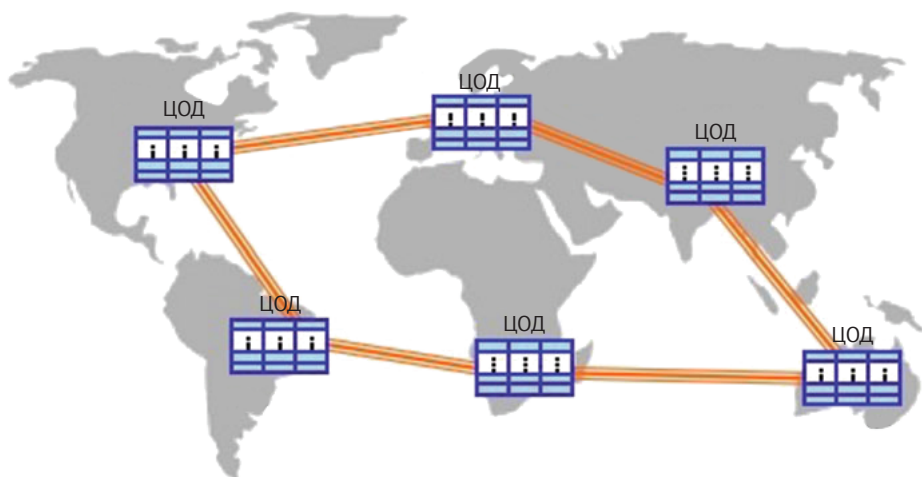


Рис. 1.1 ❖ Территориально распределенные центры обработки данных

1.1. ЧТО ТАКОЕ РАСПРЕДЕЛЕННАЯ СИСТЕМА БАЗ ДАННЫХ?

Распределенную базу данных мы определяем как набор из нескольких логически взаимосвязанных баз данных, расположенных в узлах распределенной системы. Распределенной системой управления базами данных (распределенной СУБД) мы далее называем программную систему, которая допускает управление распределенной базой данных и делает распределенность прозрачной для пользователей. Иногда, говоря «распределенная система баз данных» (распределенная СУБД), имеют в виду как распределенную базу данных, так и распределенную СУБД в нашем понимании. Мы выделяем две важные характеристики: логическую взаимосвязанность данных и их нахождение в распределенной системе.

Существование распределенной системы – важный момент. В этом контексте под *распределенной вычислительной системой* понимается несколько связанных между собой автономных обрабатывающих элементов (ОЭ). Возможности обрабатывающих элементов могут различаться, они могут быть гетерогенными, связи между ними тоже могут быть различными, но важно то, что ОЭ не имеют прямого доступа к состоянию друг друга, а могут узнать его, лишь обмениваясь сообщениями и, следовательно, неся затраты на коммуникацию. Поэтому если данные распределены, то доступ к ним и управление ими логически непротиворечивым способом требуют особого внимания со стороны распределенной СУБД.

Распределенная СУБД – это не просто «набор файлов», которые можно по отдельности хранить в каждом ОЭ распределенной системы (обычно называемом «узлом» распределенной СУБД); данные в распределенной СУБД взаимосвязаны. Мы не будем конкретизировать, что означает «взаимосвязаны», поскольку требования зависят от типа данных. Например, в случае реляционных данных различные отношения или их части могут храниться в разных узлах (подробнее об этом см. главу 2), и для ответа на запросы, выражаемые, как правило, на языке SQL, требуется выполнять операции соединения или объединения. Обычно можно определить схему таких распределенных данных. На другом полюсе находятся данные в системах NoSQL (см. главу 11), в которых возможно гораздо менее ограничительное определение взаимосвязанности; например, это могут быть вершины графа, хранящиеся в разных узлах.

Подводя итоги этому обсуждению, можно сказать, что распределенная СУБД *логически едина, но физически распределена*. Это означает, что пользователь, работающий с распределенной СУБД, воспринимает ее как единую базу данных, хотя составляющие ее данные физически находятся в разных местах.

Как уже было сказано, обычно рассматриваются два типа распределенных СУБД: территориально распределенные (или *геораспределенные*) и сосредоточенные в одном месте (одноузловые). В первом случае узлы соединены между собой глобальной сетью, для которой характерны большие задержки при передаче сообщений и более высокая частота ошибок. А во втором речь идет о системах, в которых ОЭ находятся близко друг к другу, так что

обмен сообщениями происходит гораздо быстрее (современные технологии позволяют считать задержку пренебрежимо малой) и с очень низкой частотой ошибок. Одноузловые распределенные СУБД обычно размещаются на кластерах компьютеров в одном ЦОДе и называются параллельными СУБД (их ОЭ по-английски называются «node» – в отличие от «site», а по-русски в обоих случаях употребляется термин «узел»). Выше отмечалось, что сегодня легко встретить распределенные СУБД, состоящие из нескольких одноузловых кластеров, соединенных глобальной сетью, т. е. имеет место гибридная многоузловая система. В этой книге мы в основном будем рассматривать задачи управления данными в узлах геораспределенной СУБД, а о проблемах одноузловых систем речь пойдет только в главах 8, 10 и 11, где обсуждаются параллельные СУБД, большие данные и системы NoSQL/NewSQL.

1.2. ИСТОРИЯ РАСПРЕДЕЛЕННЫХ СУБД

До появления систем баз данных в 1960-х годах каждое приложение само определяло свои данные и управляло ими (рис. 1.2). В этом режиме приложение принимало решения о структуре и методах доступа к данным и отвечало за управление файлом в системе хранения. Это приводило к значительной и неконтролируемой избыточности данных и высоким трудозатратам программистов, вынужденных заниматься управлением данными в приложениях.

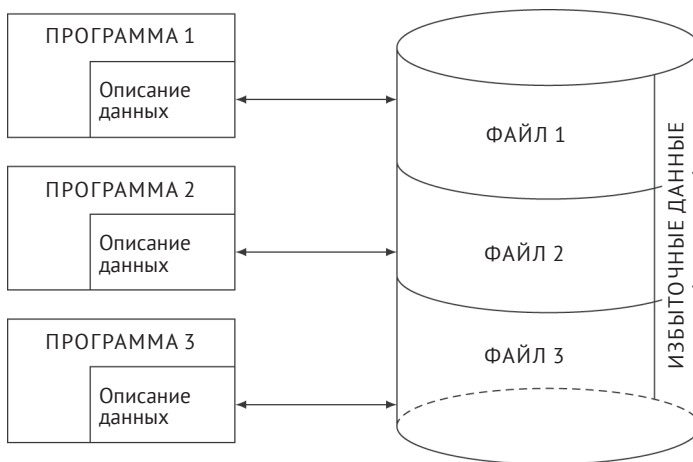


Рис. 1.2 ❖ Традиционная обработка файлов

Система баз данных позволяет определять и администрировать данные централизованно (рис. 1.3). Этот новый подход ведет к *независимости данных*, когда прикладная программа безразлична к изменению логической или физической организации данных, и наоборот. Таким образом, программисты освобождаются от ответственности за управление и сопровождение нужных им данных, а избыточность можно устранить (или уменьшить).

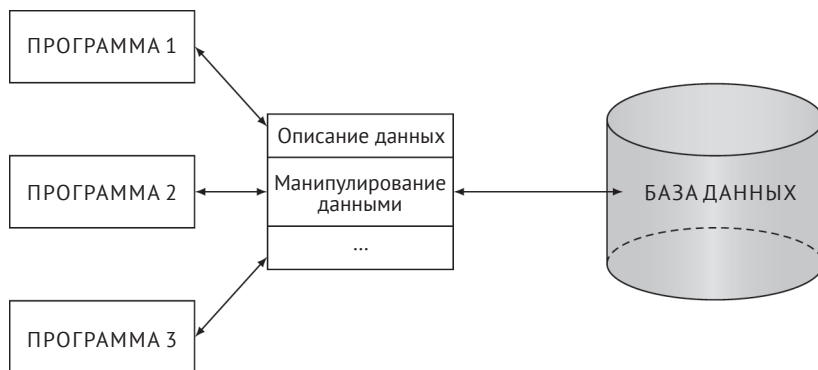


Рис. 1.3 ❖ Обработка базы данных

Одним из побудительных мотивов использования систем баз данных являлось желание объединить всю информацию о работе предприятия и предоставить интегрированный и контролируемый доступ к этим данным. Мы сознательно употребляем термин «интегрированный», а не «централизованный», потому что данные, как уже было сказано, могут размещаться на разных территориально разнесенных компьютерах. Именно в этом смысл технологии распределенных баз данных. Мы уже отмечали, что физически распределенная система может находиться в одном месте или в нескольких местах. Поэтому каждый узел на рис. 1.5 может представлять собой центр обработки данных, соединенных сетью с другими центрами. Именно распределенные среды такого типа являются типичными в настоящее время, и именно их мы будем изучать в этой книге.

С годами архитектура распределенной системы баз данных претерпела значительные изменения. Такие ранние распределенные системы баз данных, как Distributed INGRES и SDD-1, проектировались как территориально распределенные системы с очень медленными сетевыми соединениями, поэтому они всеми силами стремились оптимизировать операции, чтобы уменьшить обмен данными по сети. Это были первые *одноранговые системы* (peer-to-peer – P2P) в том смысле, что все узлы имели похожую функциональность в части управления данными. С развитием персональных компьютеров и рабочих станций стала преобладать модель распределения типа *клиент-сервер*, в которой данные переместились на тыловой сервер, а пользовательские приложения работали на фронтальных рабочих станциях. Особенно часто такие системы стали развертываться на одной территории, где можно было обеспечить более высокую скорость сети и, стало быть, более частое взаимодействие между клиентами и сервером (или серверами). В 2000-х годах произошло возрождение P2P-систем, в которых исчезло различие между клиентскими и серверными компьютерами. Современные P2P-системы отличались от ранних в нескольких важных отношениях, которые мы обсудим ниже в этой главе. Все эти архитектуры существуют и сегодня и обсуждаются в последующих главах.

Становление Всемирной паутины (или просто веба) как основной платформы для совместной работы и обмена файлами оказало громадное влияние

на исследования по управлению распределенными данными. Стало доступно гораздо больше данных, но это не были тщательно структурированные и точно определенные данные, как в типичной СУБД; они были слабо структурированы или не структурированы вовсе (т. е. какую-то структуру они имели, но определенную не на уровне схемы базы данных), их происхождение было неизвестно (т. е. данные могли быть «грязными» или ненадежными), и зачастую они были противоречивы. Ко всему прочему многие данные хранились в системах, к которым не было простого доступа (в *скрытом вебе*). Поэтому усилия по распределенному управлению данными направляются на доступ к этим данным осмысленными способами.

Это направление развития стимулировало исследования в области *интеграции баз данных* – дисциплине, которая существовала с самого начала работ по распределенным базам данных. Первоначально эти усилия были направлены на поиск способов доступа к данным в различных базах (отсюда и термины *федеративная база данных* и *мультибаза данных*), но с появлением веб-данных фокус сместился в сторону виртуальной интеграции данных разных типов (поэтому термин *интеграция данных* стал более популярным). Сейчас в моде термин *озеро данных*, который подразумевает, что все данные собираются в логически едином хранилище, из которого каждое приложение извлекает нужные ему данные. Мы обсудим интеграцию данных в главе 7, а озера данных – в главах 10 и 12.

За последние десять лет важным явлением стали облачные вычисления. Под этим понимается вычислительная модель, в которой ряд поставщиков услуг предоставляют в общее пользование разделяемые и территориально распределенные вычислительные ресурсы, так что пользователи могут арендовать их по мере необходимости. Клиенты могут взять в аренду базовую вычислительную инфраструктуру для разработки собственного программного обеспечения, а затем решить, какую операционную систему предпочитают, и создать для себя виртуальные машины (VM) со средой, в которой хотят работать, – такой подход называется *инфраструктура как услуга* (IaaS). Возможна и более развитая облачная среда, в которой в аренду сдается не только базовая инфраструктура, но и вся вычислительная платформа, на которой клиенты разрабатывают свое ПО, – это *платформа как услуга* (PaaS). Самый продвинутый вариант – когда поставщик услуг предоставляет в аренду конкретное программное обеспечение, этот подход называется *программное обеспечение как услуга* (SaaS). В последнее время начинают предлагать услуги управления распределенной базой данных в облаке как часть SaaS.

Мы дадим общий обзор всех этих архитектур в разделе 1.6.1.2, а затем обсудим их в отдельных главах более подробно.

1.3. Различные способы доставки данных

В распределенных базах данных доставка данных производится между узлами – либо от серверных узлов клиентским в ответ на запросы, либо между несколькими серверными узлами. Мы будем характеризовать варианты

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru