Посвящается нашим семьям и родителям

Содержание

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-

пределенных 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 Bondiombouy). 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 Штётер (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 Нааке (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 Бень 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.

Ватерлоо, Канада — М. Тамер Ёсу (tamer.ozsu@uwaterloo.ca)
Монпелье, Франция — Патрик Вальдуриес (patrick.valduriez@inria.fr)
Июнь 2019

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.

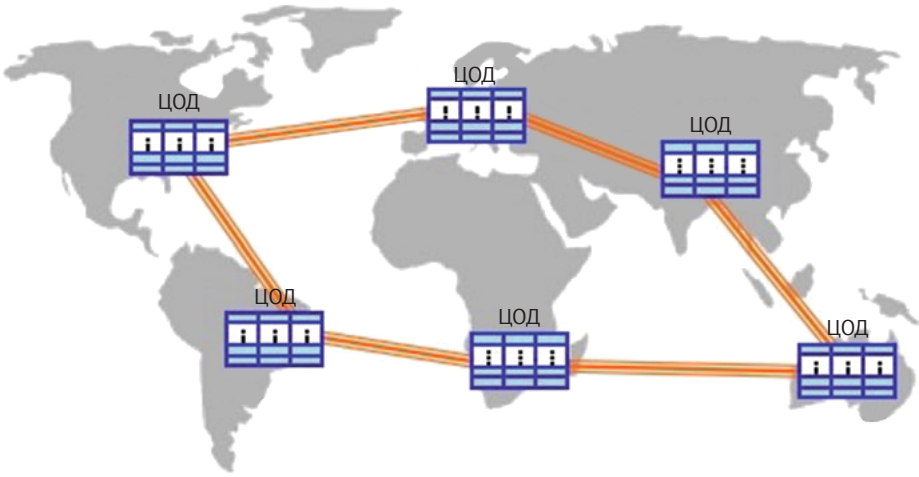

Рис. 1.1 ❖ Территориально распределенные центры обработки данных

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 *георапределенные*) 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.

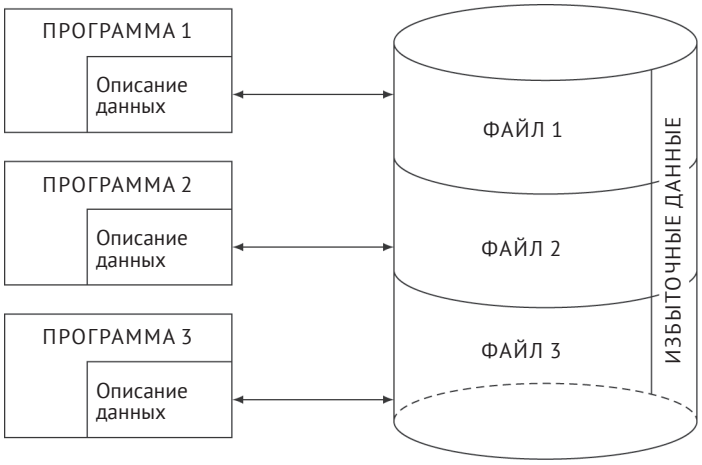

Рис. 1.2 ❖ Традиционная обработка файлов

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).

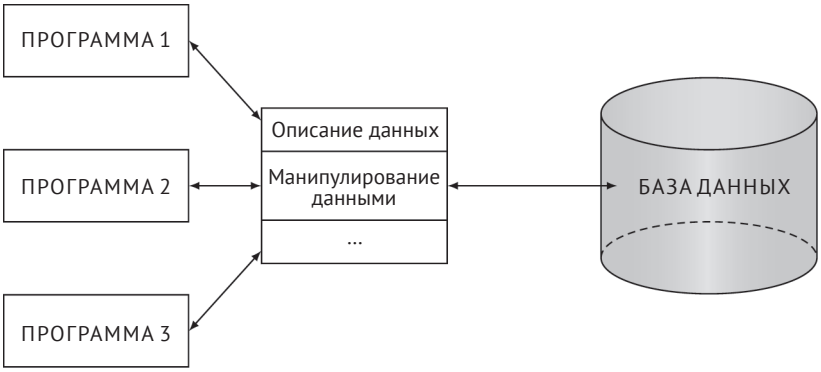

Рис. 1.3 ❖ Обработка базы даннх

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 (ВМ) 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