Содержание

Предисловие

Спросите любого, кто когда-либо присутствовал на конференции, посвященной базам данных, писал посты или вел блоги по этой теме, какой вопрос является наиболее актуальным во все времена, и вы наверняка получите один и тот же ответ – повышение эффективности. И если лекции по проектированию баз данных традиционно набирают достаточное количество посетителей, то на семинары, посвященные оптимизации БД, бывает, просто не пробиться. В чем здесь дело? Мне кажется, все очень просто – так же просто, как и основная цель оптимизации, состоящая в том, чтобы медленное сделать быстрым. Этому главным образом посвящена ежедневная профессиональная деятельность администраторов баз данных, разработчиков отчетов и бизнес-аналитиков. Скорость естественным образом преобразуется в удобство использования инструмента и быстроту принятия решений, что положительно сказывается на моральном духе коллектива и критически важных показателях организации. Да и сами разработчики, способные повысить скорость выполнения запросов и формирования отчетов, обычно не остаются в стороне и получают повышения и прибавку в зарплате.

Power BI в этом отношении ничем не отличается от любого другого инструмента бизнес-аналитики или базы данных. Одной из самых популярных причин недовольства пользователей является скорость формирования отчетов. В обычных условиях Power BI славится своей высокой производительностью даже при работе с довольно большими объемами данных. Но достаточно допустить небольшую ошибку при написании сложного вычисления или проектировании модели данных, и вы не оберетесь проблем. Будучи специалистом в области Power BI, вы должны уметь оптимально с точки зрения производительности проектировать модели данных и решать возникающие проблемы с отчетами.

Все это значительно повышает значимость книги, которую написал Бхавик. Несмотря на большую популярность темы оптимизации, я лично не видел до этого ни одной книги из этой области в Power BI. В этой книге собраны вместе советы, подсказки и приемы, которые раньше были беспорядочно разбросаны по официальной документации, блогам, курсам и статьям, и положены на огромный опыт автора в составе отдела разработки Power BI во взаимодействии с крупнейшими заказчиками. Вместо того чтобы сосредоточиться на одном аспекте оптимизации, например выражениях DAX, автор рассмотрел тему повышения эффективности Power BI действительно многогранно и всесторонне. В результате мы получили бесценный ресурс, способный стать краеугольным камнем на пути совершенствования навыков в деле оптимизации проектов на базе Power BI. Строго следуйте всем советам из этой книги и воплощайте их в жизнь!

Кристофер Уэбб,
главный администратор команды Power BI CAT,
13-кратный обладатель статуса MVP
и автор множества книг в области SSAS и Power BI

Об авторе

Бхавик Мерчант (Bhavik Merchant) обладает 18-летним опытом работы в области бизнес-аналитики и занимает пост руководителя отдела продуктовой аналитики в Salesforce. До этого работал в Microsoft сначала в роли архитектора облачных решений, а затем в статусе продуктового менеджера в проектной группе Power BI. В отделе Power BI Бхавик возглавлял программу клиентских исследований, отвечая за стратегию и техническую структуру предоставления клиентам информации о производительности системы. До Microsoft много лет работал консультантом BI-систем в отделе корпоративных клиентов. Проводил технические и теоретические тренинги в области повышения эффективности Power BI для партнеров Microsoft по всему миру.

О редакторах

Суреш Датла (Suresh Datla) работает в IT-индустрии более 20 лет и обладает большим опытом в области бизнеса и технологий. Он является разработчиком, консультантом, популяризатором и тренером по Power BI. С момента появления на рынке Azure и Power Platform тесно работает с этими системами, а также является частью проекта Microsoft по разработке и внедрению вертикальных решений. Суреш неоднократно выступал на мероприятиях от Microsoft по темам Power Platform, Power BI, Power BI Premium, безопасности и эффективности. Каждый месяц организовывает форум по Power Platform в Южной Калифорнии и свято верит в то, что своим успехом эта платформа всецело обязана квалифицированному сообществу. Суреш является директором компании Synergis Consulting, возглавляя группы архитектуры данных, разработчиков и инженеров.

Вишванат Музумдар (Vishwanath Muzumdar) имеет более чем 8-летний опыт работы в сфере информационных технологий и бизнес-аналитики. Специализируется на создании визуальных отчетов для клиентов. Своей целью видит применение управленческих и аналитических навыков в сфере инструментов отчетности Microsoft Power BI для помощи компании в достижении финансовых успехов.

Введение

Начать выстраивать аналитические решения с помощью Power BI очень несложно. После этого проект может жить собственной жизнью, набирая популярность и повышая объем используемых данных. Однако если не запланировать такой рост изначально, вы наверняка в какой-то момент столкнетесь с проблемами. Эта книга поможет вам провести мероприятия по оптимизации всех без исключения слоев Power BI, начиная с рабочей области отчета и заканчивая моделированием данных, их преобразованием, хранением и архитектурой.

Разработчики и архитекторы, работающие с Power BI, смогут применить полученные из этой книги знания на практике на всех стадиях жизненного цикла своих решений. Книга, которую вы держите в руках, – это не просто сборник советов и приемов по оптимизации своих проектов, но и полное структурированное руководство для обнаружения узких мест и их устранения.

Изучив все приведенные практики и примеры, вы научитесь определять распространенные ошибки на этапе проектирования данных, приводящие к снижению эффективности решения и расходованию лишней памяти. Мы рассмотрим все настройки, которые могут негативно сказываться на производительности. Вместе мы пройдем по всем слоям типичного решения Power BI и узнаем, что необходимо сделать, чтобы при масштабировании проекта не страдало его быстродействие. Начнем мы со слоя данных и постепенно поднимемся до уровня дизайна отчетов. Попутно мы рассмотрим варианты лицензирования Power BI Premium, включая процесс планирования загрузки и нагрузочное тестирование, и поговорим о службах Azure, позволяющих обеспечить дополнительное масштабирование.

Прочитав книгу, вы сможете поддерживать решения Power BI любой степени сложности с минимальными усилиями. Вдобавок вы научитесь использовать сторонние программные продукты для обнаружения проблем с производительностью.

Для кого эта книга

Книга, которую вы начинаете читать, предназначена для аналитиков данных, разработчиков в области бизнес-аналитики и специалистов по работе с Power BI. Она будет полезна тем, кто хочет создавать решения на базе Power BI, способные масштабироваться в отношении объема данных и количества пользователей без потери эффективности. Также книга поможет идентифицировать и устранить узкие места, влияющие на производительность решения. Для понимания всех концепций, описанных в этой книге, вам потребуется базовое знание Power BI и всех его компонентов.

Структура книги

Глава 1 «**Постановка целей и определение проблемных областей».** В этой главе мы рассмотрим решение на базе Power BI в виде потока данных от различных источников к потребителям в обобщенном виде. Мы посмотрим, как могут храниться и перемещаться данные в Power BI на пути к конечным потребителям. В большинстве случаев решения относительно архитектуры проекта, принятые на ранней стадии, бывает трудно и дорого отменить или изменить. Именно поэтому необходимо на самом старте проекта правильно оценивать нагрузку и планировать разработку, исходя из нее.

Глава 2 «**Обзор архитектуры и конфигурации Power BI».** Из этой главы вы узнаете, как можно улучшить производительность и снизить время ожидания получения информации. Здесь мы также расскажем о режимах хранения данных в Power BI и их перемещении в модель данных, поскольку решения, принятые на этом этапе, могут влиять на объемы и актуальность исходных данных. Кроме того, мы рассмотрим разные варианты развертывания шлюзов Power BI, обычно используемых для подключения к внешним источникам данных. Важность этой темы обусловлена тем, что пользователям зачастую требуется оперировать одновременно актуальными и историческими данными, объем которых ничем не ограничен.

Глава 3 «**Оптимизация DirectQuery».** В третьей главе книги мы познакомимся с режимом хранения DirectQuery, полагающимся на внешние источники данных. Этот режим, как правило, используется в организациях при наличии больших объемов данных. Источники DirectQuery зачастую не предназначены для аналитических запросов, что негативно сказывается на быстродействии отчетов и операций обновления данных. Мы рассмотрим методы оптимизации как в отношении Power BI, так и применительно к внешним источникам, что позволит повысить эффективность запросов.

Глава 4 «**Анализ логов и метрик».** В этой главе мы поговорим о том, что быстродействие отчетов может быть улучшено только в случае ее объективной оценки. Таким образом, здесь мы узнаем, где можно взять данные о производительности и как по ним определить наличие узких мест в системе. Будут рассмотрены встроенные и внешние инструменты для мониторинга показателей эффективности, а также даны полезные рекомендации по проведению такого анализа.

Глава 5 «**Анализатор производительности».** Здесь мы поговорим об одном из самых простых способов отслеживания временных задержек при формировании отчетов. Мы воспользуемся инструментом **Анализатор производительности**, служащим для проведения подробного анализа действий пользователей с детализацией до визуальных элементов. Мы выполним расширенный обзор всех возможностей, опишем все метрики и продемонстрируем процесс анализа на примере.

Глава 6 «**Внешние инструменты».** Данная глава будет посвящена сторонним инструментам, способным помочь при анализе производительности решений. Мы рассмотрим типичные сценарии использования таких инструментов с подключением к Power BI, сбором ключевых показателей эффективности и их подробным анализом.

Глава 7 «**Общие принципы управления производительностью».** В этой главе мы расскажем о том, что метрики и инструменты, описанные в предшествующих главах, по сути, являются строительными блоками общей системы управления показателями эффективности. При этом успех более вероятен при внедрении структурированного и воспроизводимого подхода к построению образа мышления на основе показателей эффективности на всех стадиях жизненного цикла решения в Power BI. Здесь мы приведем советы по процессу управления данными, которые помогут избежать проблем с масштабированием для новой информации и предотвратить ухудшение качества данных для имеющейся. Мы также обсудим типичные роли в рамках аналитического проекта любого уровня – от самостоятельного до управляемого из единого центра – и расскажем об их функциях в деле повышения эффективности решения.

Глава 8 «**Загрузка, преобразование и обновление данных».** Здесь мы поговорим о важнейшей роли периодического обновления данных для любой аналитической системы, и в Power BI это применимо к наборам данных в режиме Import. Операции по обновлению данных в этом режиме являются одними из наиболее затратных в отношении нагрузки на центральный процессор и память, и они могут повлечь серьезные задержки и даже отказы, особенно при работе с объемными наборами данных. В результате пользователи могут остаться без обновлений, процесс разработки замедлится, а ресурсы будут постоянно подвергаться огромным нагрузкам. Чтобы избежать этих проблем, требуется при преобразовании данных уделить особое внимание вопросам производительности.

Глава 9 «**Разработка отчетов и дашбордов».** В этой главе мы поговорим о вершине айсберга любого решения Power BI, представляющей собой отчеты и дашборды, с которыми по большей части взаимодействует пользователь. Независимо от своего визуального представления, этот слой Power BI по своей сути является приложением JavaScript, запущенным в браузере. Здесь мы коснемся ключевых приемов, позволяющих оптимизировать вывод визуального слоя, включая срезы и фильтрацию. Также мы поговорим о постраничных отчетах, которые ведут себя отлично от интерактивных и обладают своими особенностями в отношении оптимизации.

Глава 10 «**Моделирование данных и безопасность на уровне строк».** Здесь мы подробно поговорим о наборах данных Power BI, в которых хранится исходная информация после преобразования и откуда извлекается для анализа. Таким образом, это наиболее критичная область любого решения на базе Power BI, лежащая в его основе. В то же время Power BI обладает достаточным арсеналом возможностей, которые можно применить в процессе моделирования данных. Некоторые решения способны облегчить процесс разработки ценой снижения производительности запросов и/или увеличения объема обрабатываемых данных. В этой главе мы дадим полезные советы по моделированию данных, снижению объемов задействованной информации и ускорению работы связей. В заключение коснемся темы оптимизации безопасности на уровне строк.

Глава 11 «**Улучшаем DAX».** В данной главе мы посмотрим, как формулы DAX позволяют разработчику расширить функционал модели данных. При

этом одного и того же результата можно добиться с применением разных формул, и не все они будут одинаково эффективными в конкретных условиях. Мы перечислим основные проблемы, связанные с формулами DAX, и научимся писать код более эффективно.

Глава 12 «Шаблоны работы с большими данными». Здесь мы узнаем, как постоянный рост объема данных в компании способен привести к серьезным проблемам. Даже с применением инновационных технологий сжатия данных, используемых в Power BI, бывает затруднительно за приемлемое время выполнять загрузку нужных нам наборов данных в режиме Import. И эта проблема усугубляется необходимостью параллельно поддерживать работу в системе сотен и тысяч пользователей. В этой главе мы рассмотрим способы борьбы с подобными проблемами, включая использование лицензирования Power BI Premium, технологий Azure, а также составных моделей и агрегатов.

Глава 13 «Оптимизация емкостей Premium и Embedded». В этой главе мы обсудим предоставляемые в рамках лицензии Power BI Premium выделенные емкости с менее строгими ограничениями, а также другие возможности, включая постраничные отчеты и службы искусственного интеллекта. Кроме того, мы поговорим о втором поколении Premium (Gen2) и узнаем, как при наличии такой подписки система справляется с повышенной нагрузкой и как работает автоматическое масштабирование. Попутно мы коснемся настроек, которые могут позволить повысить производительность системы. Мы научимся правильно планировать объемы данных и проводить нагрузочные тесты. Также мы посмотрим, как можно использовать приложение Capacity Metrics для поиска и решения проблем с нагрузкой на емкость.

Глава 14 «Встраивание в приложения». В заключительной главе книги мы посмотрим, как можно встраивать содержимое Power BI в пользовательские веб-приложения для интегрирования информации с данными из других источников. В этом случае приложение размещается на стороннем сервере при помощи вызовов API, что накладывает дополнительные ограничения. Мы поговорим о том, как можно организовать этот процесс наиболее эффективно.

Как извлечь максимум из книги

К некоторым главам этой книги прилагаются файлы с примерами, которые можно открыть с помощью Power BI Desktop. Это поможет вам лучше понять описываемые концепции и приемы. В основном в примерах показаны ситуации до и после внесенных изменений. Вам не обязательно проверять все представленные теории на примерах, но они действительно могут помочь в их освоении.

Программное обеспечение, использованное при написании книги: ОС Windows, Power BI Desktop, DAX Studio 2.17.3, Tabular Editor 3, Power BI Helper 12.0.

Мы советуем вам всегда работать с самой свежей версией Power BI Desktop, следуя их ежемесячным обновлениям.

Если вы читаете электронную версию книги, мы советуем вам вводить код самостоятельно или копировать его из репозитория книги на GitHub (ссылка

будет дана в следующем разделе). Это поможет вам избежать ошибок при копировании скриптов из книги.

Сопроводительные файлы

Файлы с примерами можно загрузить из репозитория книги на GitHub по адресу https://github.com/PacktPublishing/Microsoft-Power-BI-Performance-Best-Practices. Все возможные обновления будут появляться там же.

Также вы можете загрузить текущую версию файлов с сайта издательства по адресу www.dmkpress.com на странице с описанием данной книги.

Цветные изображения

По следующей ссылке вы можете скачать в виде PDF все рисунки и диаграммы, использованные в книге: https://static.packt-cdn.com/downloads/9781801076449_ColorImages.pdf.

Условные обозначения

На протяжении книги мы будем использовать следующие условные обозначения и шрифты.

`Код в тексте`: так в тексте книги мы будем обозначать код, имена таблиц баз данных, имена папок, файлов, расширения файлов, пути, ссылки, пользовательский ввод. Пример: «Просто скопируйте приведенный ниже код, выполните его, после чего перезапустите `TabularEditor`, чтобы применились новые правила».

Блоки кода будут выделены следующим образом:

```
System.Net.WebClient w = new System.Net.WebClient();
string path = System.Environment.GetFolderPath(System.Environment.SpecialFolder.
LocalApplicationData);
string url = "https://raw.githubusercontent.com/microsoft/Analysis-Services/master/
BestPracticeRules/BPARules.json";
string downloadLoc = path+@"\TabularEditor\BPARules.json";
w.DownloadFile(url, downloadLoc);
```

Новые термины, важные слова и текст, который вы видите на экране, будут выделены жирным шрифтом, например: «Раздел **Workloads** содержит настройки, связанные с производительностью».

Важные примечания будут выводиться так.

Советы будут выводиться так.

Примечания будут выводиться так.

Часть I

Архитектура, узкие места и целевые показатели производительности

В этой вводной части мы дадим высокоуровневый обзор архитектуры Power BI и определим области, в которых на производительность можно влиять посредством проектных решений. После прочтения данной части вы будете понимать, как устанавливать реалистичные целевые показатели производительности.

Содержание этой части:

- глава 1 «Постановка целей и определение проблемных областей»;
- глава 2 «Обзор архитектуры и конфигурации Power BI»;
- глава 3 «Оптимизация DirectQuery».

Глава 1

Постановка целей и определение проблемных областей

При анализе производительности аналитических решений многие считают важнейшим показателем быстродействие системы формирования отчетности. По большей части это так и есть, поскольку практически все пользователи – от операторов до управляющих – взаимодействуют с системой именно посредством отчетов как главной визуальной составляющей. Однако скоро вы узнаете, что существуют и другие не менее важные области для применения оптимизации, если смотреть на ситуацию в целом. К примеру, ускорение подсистемы формирования отчетов может не дать ожидаемых результатов, если исходный набор данных, лежащий в основе отчета, долго обновляется или выдает ошибки из-за достигнутых ограничений выделенных ресурсов. В итоге актуальные свежие данные могут просто не поспевать за великолепными и быстро формирующимися отчетами.

Автор книги, которую вы держите в руках, испытал на себе последствия снижения быстродействия системы отчетов. Как-то раз в одной крупной коммунальной компании предприняли попытку миграции с одной системы формирования отчетности на другую, от стороннего поставщика. Несмотря на превосходство новой системы в техническом и функциональном планах, разработчики попытались напрямую перенести в нее функционал старых отчетов. В результате это решение привело к существенному снижению быстродействия отчетов. Были потрачены миллионы на лицензирование и консультации с новым поставщиком, но большинство пользователей просто отказывались переходить на новую систему из-за ее медлительности. Этот случай мы привели в качестве демонстрации возможных последствий того, что может произойти, если изначально правильно не заложить фактор производительности в аналитическое решение.

В этой главе вы начнете свое путешествие в мир оптимизации решений на базе Microsoft Power BI. В качестве введения в полный спектр управления производительностью мы рассмотрим решения Power BI в образе потока данных от различных источников в консолидированном виде и их представления аналитикам данных и прочим пользователям, работающим с инфор-

мацией. Мы посмотрим, как данные могут храниться в Power BI и какой путь преодолевать по дороге к конечному пользователю. Многие архитектурные и проектные решения, принятые на ранней стадии становления проекта, бывает очень проблематично и дорого отменять или изменять впоследствии. В связи с этим возрастает важность всесторонней предварительной оценки возможных последствий принимаемых решений и использования подхода на основе данных к выбираемой стратегии на самом старте.

При оценке эффективности системы разработчики зачастую недооценивают или вовсе упускают из виду процесс установки *целевых показателей производительности* (performance targets). А как без этого определить, какого результата вы в итоге добились? Давайте начнем с теоретической части описания целей, после чего перейдем к техническим аспектам реализации.

Темы, которые будут рассмотрены в этой главе:

- определение уровня производительности;
- области с возможными замедлениями;
- решения, влияющие на производительность.

Определение уровня производительности

С появлением сверхбыстрых компьютеров и средств распределенного вычисления пользователи и заказчики аналитических решений вполне обоснованно стали ожидать от них впечатляющего быстродействия, что бывает критически важно для принятия серьезных бизнес-решений. Поставщики инструментов бизнес-аналитики откликнулись на эти ожидания упоминаниями во всех своих рекламных проспектах потрясающей скорости работы предлагаемых ими решений. В результате сегодня с трудом можно найти пользователя, который воспринял бы действительно быстро формирующиеся отчеты или вовремя обновляющиеся данные как нечто необычное и поражающее воображение – скорее, как само собой разумеющееся. И наоборот, любая задержка в процессе формирования отчетов приводит к резкой негативной реакции с его стороны и жалобам во все возможные инстанции. Если такие проблемы начинают носить масштабный характер, критике начинает подвергаться как сама платформа, такая как Power BI, так и техническая команда, занимающаяся разработкой конкретного решения. В худшем случае пользователи могут отказаться работать в предлагаемом программном комплексе, в результате чего руководство может принять решение о смене платформы. Решение состоит в том, чтобы на самых ранних стадиях проекта задуматься о быстродействии строящегося проекта, поскольку исправить возникшие проблемы с производительностью при выходе на рабочие мощности с привлечением тысяч пользователей может оказаться очень сложно, если не невозможно.

Показатели производительности отчетов

Сегодня большинство решений в области бизнес-аналитики существуют в виде веб-интерфейса. При этом работа с отчетами обычно не ограни-

чивается одним лишь их формированием – пользователи с ними активно взаимодействуют. Применительно к Power BI это означает открытие отчета и дальнейшую интерактивную работу с фильтрами, срезами и визуальными элементами, детализацию до нужных уровней и переключение между страницами как непосредственно, так и с помощью закладок. Каждое взаимодействие пользователя с отчетом имеет определенное намерение, и нельзя разрывать эту связь. В нашей отрасли бытует высказывание о том, что *аналитика должна производиться со скоростью мысли*. Этот опыт и связанные с ним ожидания очень напоминают навигацию по обычному веб-сайту или взаимодействие с программным обеспечением в интернете.

Таким образом, при определении показателей производительности для аналитической системы можно воспользоваться многими наработками, принятыми в веб-студиях и используемыми на протяжении последних двух-трех десятилетий, – это не так сложно. В своей статье от 2004 года профессор Фиона На (Nah, F.) определила показатель *приемлемого времени ожидания* (tolerable wait time – TWT) для веб-пользователей. Этот показатель описывает время, которое пользователь сайта готов ждать, перед тем как закрыть веб-страницу. В своей статье профессор привела многочисленные исследования прошлых лет, ориентированные на определение пороговых временных отметок, по достижении которых у пользователя истекает терпение и появляется негатив. Из этих исследований можно сделать вывод о том, что в хорошем отчете Power BI должна полностью загружаться страница или появляться результат интерактивного взаимодействия в идеале в течение четырех секунд, а в большинстве случаев – не позже, чем через 12 секунд. При этом измерение всегда стоит производить с точки зрения пользователя, т. е. с момента вызова им отчета (например, нажатия на ссылку на веб-портале Power BI) и до завершения полной отрисовки отчета на экране.

Установка реалистичных целевых показателей производительности

Теперь, когда у нас есть руководство по установке целевых показателей на основе исследований, нам нужно применить его на практике. Распространенной ошибкой является установка единого целевого показателя для всех отчетов и ожидание, что он будет выполняться при каждом взаимодействии пользователя с любым из них. Недостаток такого подхода заключается в том, что даже хорошо спроектированная и оптимизированная система может оказаться слишком сложной для удовлетворения оптимистично установленной цели. К примеру, при наличии очень большого набора данных (десятки гигабайт) и сложного вложенного выражения DAX, результат которого отображается в табличном визуальном элементе (**Table**) с несколькими уровнями гранулярности, вам будет никак не уложиться в рамки, приемлемые для небольшого датасета с простыми агрегациями в виде суммирования и отображением в виде карточек (**Card**).

Таким образом, по причине изменчивости сложности решений и других факторов, не зависящих от разработчика (к примеру, мощности компьютера

пользователя или используемого им браузера), стоит рассматривать целевые показатели производительности в терминах *типичного опыта использования* (typical user experience) и предполагать, что есть как ожидания, так и выбросы. В результате *целевые метрики* должны вычисляться с расчетом на большинство пользователей. Мы рекомендуем устанавливать целевые показатели с использованием 90-го *процентиля* для загрузки отчета или интерактивного взаимодействия с ним пользователя. Часто такая метрика называется *P90*. С учетом приведенных выше исследований целевая метрика P90 по загрузке отчета должна составлять 10 секунд или меньше. Это означает, что 90 % загрузок отчета должны укладываться в интервал до 10 секунд включительно.

В то же время одного только показателя P90 будет недостаточно, и мы подробно будет говорить об этом в главе 7. На данный момент мы должны учитывать, что решения могут быть разной степени сложности, в связи с чем рекомендуется устанавливать набор метрик в зависимости от характера отчета и его допусков по ожиданию. На рис. 1.1 показан пример таблицы с целевыми показателями, которую можно адаптировать под конкретные нужды организации.

	Обычный отчет	Сложный отчет
Целевая метрика P90	Меньше 10 секунд	Меньше 25 секунд

Рис. 1.1 ❖ Пример целевых метрик для отчетов в Power BI

Теперь взглянем на Power BI в целом, чтобы лучше понимать, в каких именно областях необходимо внедрять оптимизацию.

Области с возможными замедлениями

На следующем шаге нашего процесса улучшения производительности мы должны понять, на что именно расходуется время. По своей сути любое решение Power BI призвано в конечном счете представлять информацию пользователю в удобном для него виде. Таким образом, само решение можно воспринимать как поток данных, берущих свое начало в определенных источниках, преодолевающих различные системные компоненты Power BI и достигающих компьютера или мобильного телефона конечного пользователя. Упрощенная схема типичного решения Power BI показана на рис. 1.2.

Теперь давайте поговорим о различных составных частях типичного решения на базе Power BI, чтобы лучше разобраться, какую роль каждая из них играет в отношении возможных проблем с производительностью. Подробное рассмотрение некоторых из этих частей мы оставим на вторую главу книги.

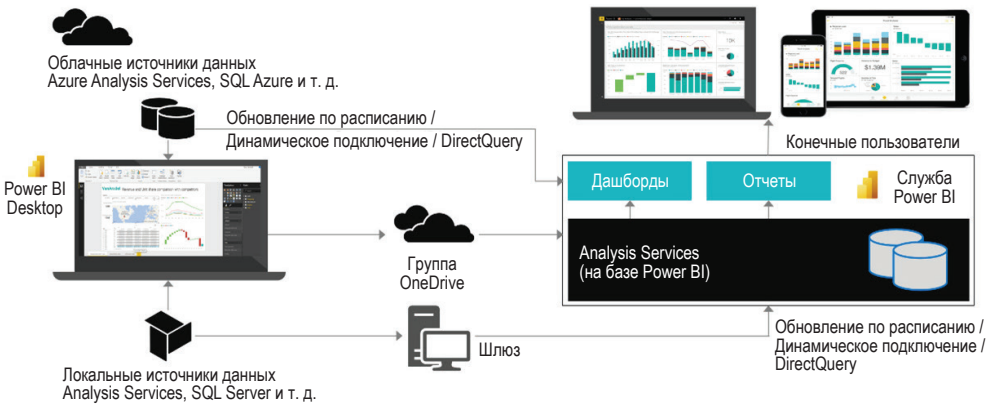

Рис. 1.2 ❖ Решение на базе Power BI в упрощенном виде

Подключение к источникам данных

На рис. 1.3 выделены области решения, на которых сказываются проблемы с источниками данных и методами подключения к ним.

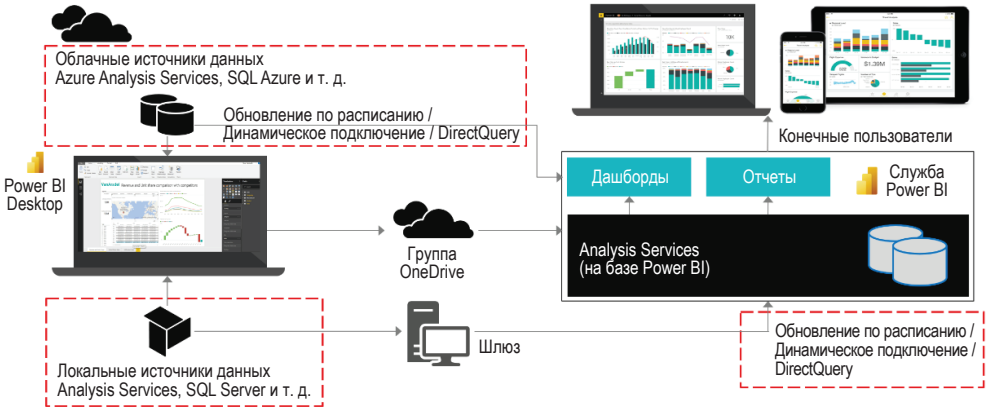

Рис. 1.3 ❖ Области, чувствительные к проблемам с подключениями и источниками данных

Режим Import

При использовании наборов данных в *режиме* ***Import*** разработчики могут испытывать проблемы с задержками от пользовательского интерфейса при работе с Power Query или языком запросов M в Power BI Desktop. В исключительных случаях это может привести к увеличению сроков разработки операций преобразования данных с часов до дней. После развертывания решения проблемы в этой области могут негативно сказываться на времени

обновления данных и даже приводить к отказам. В службе Power BI принято ограничение на обновление данных, равное двум часам, тогда как при наличии лицензии Power BI Premium это ограничение увеличивается до пяти часов. При превышении этих лимитов любое обновление будет отменено системой.

Режим DirectQuery

При использовании *режима* ***DirectQuery*** данные физически остаются в источнике, что предполагает необходимость их извлечения и обработки в Power BI практически при каждом взаимодействии пользователя с системой. При применении этого режима проблемы обычно возникают в отношении скорости формирования отчетов пользователем. Визуальные элементы будут загружаться дольше, пользователи будут нервничать, прерывать загрузку и пытаться взаимодействовать с другими элементами или менять фильтры. Это само по себе может привести к увеличению количества отправляемых запросов, что еще больше замедлит процесс формирования отчетов за счет дополнительных операций загрузки из внешних источников.

Режим Live connection

Изначально *режим* ***Live connection*** относился исключительно к подключениям к внешним ресурсам Analysis Services, которые могли быть как облачными (Azure Analysis Services), так и локальными (SQL Server Analysis Services). Позже, с появлением *общих наборов данных* (shared datasets) и возможности строить отчеты в Power BI Desktop на основании опубликованных наборов данных в службе Power BI, этот режим был расширен. Поскольку исходные наборы данных могут быть как в режиме Import, так и в режиме DirectQuery, опыт работы с ними может отличаться, как описано в предыдущих разделах.

Шлюз Power BI

Шлюз (gateway) Power BI – это промежуточный компонент, использующийся для подключения к внешним источникам данных. Обычно он располагается в той же физической или виртуальной сети и обеспечивает безопасное исходящее соединение с Power BI, по которому могут передаваться данные для отчетов и обновлений.

Но функции шлюза не ограничиваются передачей данных – это распространенное и очень большое заблуждение. Помимо обеспечения защищенного соединения с источниками данных, шлюз содержит свой *движок обработки* (*mashup engine*), выполняющий преобразование исходных данных и их сжатие перед отправкой в службу Power BI. При отсутствии оптимизации шлюза могут возникать задержки с обновлением данных вплоть до сбоев, замедление взаимодействий пользователей с отчетами и отказы загрузки визуальных элементов из-за превышения времени выполнения запросов.

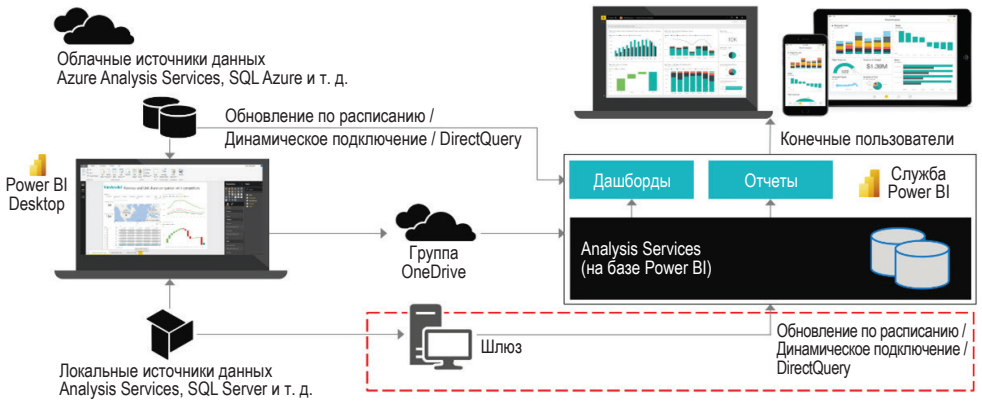

Рис. 1.4 ❖ Шлюз Power BI

Сетевая задержка

Сетевая задержка (*network latency*) выражается во времени, необходимом для передачи порции данных из одной точки сети в другую. Этот показатель измеряется в миллисекундах, и обычно для этого используется утилита ***ping***. При помощи нее фиксируется время, затраченное на отправку небольшого пакета информации в место назначения и получения ответа о его успешной доставке адресату. Если это время исчисляется секундами, вас могут ждать серьезные проблемы. Основными факторами, влияющими на сетевую задержку, являются географическое расстояние между точками, количество *транзитных участков*, или так называемых *прыжков* (*hops*), на пути и загруженность сети в целом.

На рис. 1.5 показаны пути, по которым данные могут перемещаться в Power BI. Стоит отметить, что для каждой стрелки может быть характерна своя сетевая задержка, а значит, разные пользователи могут ощущать разную скорость передачи данных при работе с системой.

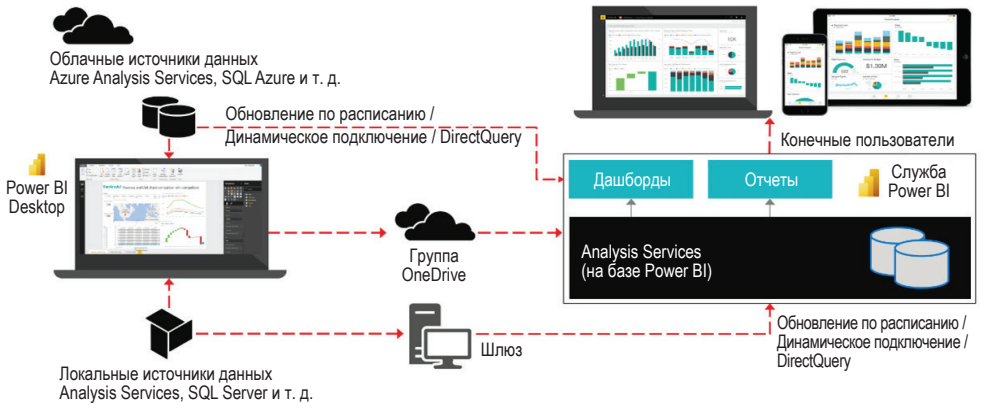

Рис. 1.5 ❖ Перемещение данных по сети

Наибольшую задержку пользователи могут испытывать при интерактивном взаимодействии с отчетами. В первую очередь это будет заметно тем, кто работает с отчетами, состоящими из множества визуальных элементов, а значит, посылающих большое количество запросов, каждый из которых должен быть обработан, а данные должны быть возвращены по сети.

Служба Power BI

Служба Power BI (*Power BI service*), выделенная на рис. 1.6, является важнейшей составляющей любого решения на базе Power BI. Системные компоненты службы в большинстве своем не поддаются управлению со стороны разработчиков и пользователей. Вместо этого их стабильность и быстродействие контролируются компанией Microsoft. Исключениями являются емкости Power BI Premium и Power BI Embedded, инфраструктура которых по-прежнему контролируется Microsoft, но администраторы организации имеют доступ к управлению выделенными им емкостями. Подробно об этом мы будем говорить в главе 13.

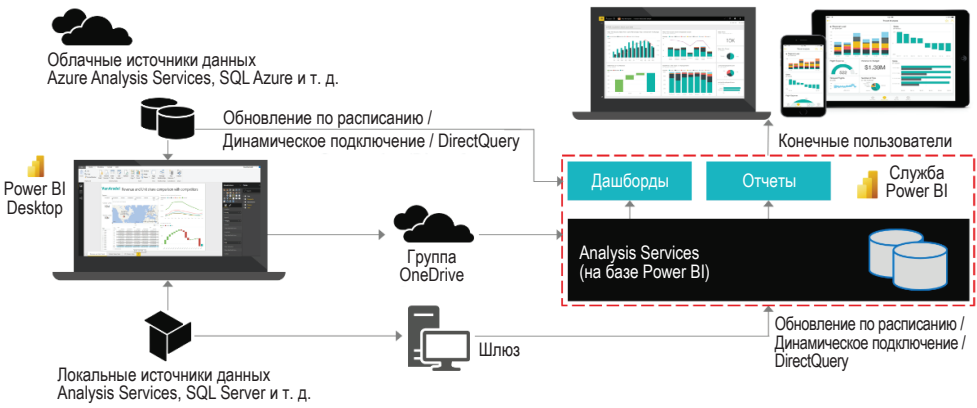

Рис. 1.6 ❖ Служба Power BI

Основным компонентом службы Power BI, которым вы можете управлять, является движок Analysis Services, лежащий в основе решения на базе Power BI. Даже при должном контроле за работой службы Power BI со стороны Microsoft неправильные решения, принятые в области моделирования данных в Analysis Services, или неоптимальные вычисления DAX могут приводить к значительному увеличению объема наборов данных, используемой памяти и, как следствие, замедлению выполняемых запросов. В результате обычно страдает быстродействие отчетов. При применении емкостей Premium/Embedded проблемы с Analysis Services могут оказывать экспоненциальное воздействие на производительность системы из-за влияния на множество наборов данных в емкости.

В заключительном разделе этой главы мы перечислим области Power BI, в которых можно добиться улучшений за счет использования разных шаб-

лонов разработки. Выбор, сделанный в этих областях, может оказывать существенное влияние на производительность.

Решения, влияющие на производительность

Каждый компонент Power BI в отдельности может быть оптимизирован для достижения лучшей производительности общего решения, и ниже мы перечислим области, с которых стоит начинать свой путь оптимизатора:

- **неправильное использование режимов DirectQuery/Import**: решения, принимаемые в отношении режимов хранения данных, оказывают влияние на соотношение между размером модели / временем ее обновления и актуальностью данных / интерактивными возможностями отчетов;
- **разработка в Power Query**: решения, принимаемые на этом слое, могут помешать использованию нативных возможностей, которыми наделен источник данных, в результате чего дополнительная нагрузка будет ложиться на внутренний движок обработки Power Query;
- **моделирование данных**: решения в этой области могут негативно сказываться на объеме модели данных и используемой памяти, а также приводить к задействованию дополнительных вычислительных ресурсов, что не может не отразиться на удобстве использования решения;
- **неэффективные вычисления DAX**: ошибки в этой области могут не позволить использовать высокоэффективный *движок хранилища* (Storage Engine) VertiPaq, вместо которого нагрузка ляжет на *движок формул* (Formula Engine);
- **сложные или неэффективные настройки безопасности на уровне строк**: промахи здесь могут приводить к необходимости производить достаточно интенсивные вычисления для определения того, какие строки должны быть доступны пользователю;
- **плохо спроектированные отчеты**: неверные решения в этой области могут приводить к повышенной нагрузке на устройства конечного пользователя;
- **задержка сети или доступа к данным**: неправильно выбранная стратегия в этой части может разнести данные и пользователей дальше друг от друга, чем это возможно.

Теперь, когда вы познакомились со всеми высокоуровневыми областями решения на базе Power BI, доступными для оптимизации, пришло время подвести итоги этой главы.

Заключение

Как вы узнали из этой главы, интерактивное взаимодействие с аналитическими отчетами очень напоминает работу с обычными веб-приложениями, в связи с чем при определении степени удовлетворенности пользователей

быстродействием и удобством системы можно опираться на уже известные принципы, разработанные для интернет-серфинга. Исследования в этой области показали, что в идеале процесс формирования отчетов должен укладываться в 4 с, тогда как превышение отметки в 10–12 с может привести к плохо скрываемому пользователями недовольству быстродействием отчетов.

В процессе оптимизации системы необходимо устанавливать целевые показатели производительности и быть готовыми к присутствию выбросов в рамках 90-го процентиля. При наличии очень сложных отчетов нужно предусмотреть набор метрик производительности, который будет учитывать характер отчета.

Важно помнить, что каждый компонент Power BI и даже сети, по которой бегают данные, вносит свой вклад в быстродействие системы в целом. В связи с этим проблемы с производительностью не стоит пытаться решать изолированно – например, путем оптимизации только отчетов. Вместо этого данный процесс может потребовать координации усилий всей команды и сторонних поставщиков, особенно это касается крупных организаций.

В следующей главе мы подробнее расскажем о работе внутреннего движка хранилища VertiPaq в Power BI и посмотрим, как можно оптимизировать хранение информации. Также мы коснемся вопросов оптимизации шлюза и дадим важные советы, которые помогут убедиться в том, что эта область не является узким местом в отношении производительности системы.