Как не утопить ваши данные в болоте

АВТОР
АЛЕКСЕЙ КАТКЕВИЧ,
архитектор Департамента разработки компании «Инфосистемы Джет»

ЕЩЕ В 2014 Г. В GARTNER ВЫПУСТИЛИ СТАТЬЮ, ОЗАГЛАВЛЕННУЮ ФРАЗОЙ «BEWARE OF THE DATA LAKE FALLACY», И ПИСАТЬ НА ЭТУ ТЕМУ В 2018 Г. ПРОСТО НЕПРИЛИЧНО. НО ПРАКТИКА ГОВОРИТ ОБ ОБРАТНОМ: ВСЕ БОЛЬШЕ И БОЛЬШЕ ЗАКАЗЧИКОВ ПРИХОДИТ С ИДЕЙ ПОСТРОИТЬ ЕДИНОЕ ХРАНИЛИЩЕ, ДА ЕЩЕ НА НОВЫХ ТЕХНОЛОГИЯХ.

Как известно, «опыт — сын ошибок трудных», и он тем ценнее, чем сильнее вас ударил черенок от граблей, на которые вы наступили. Но когда речь идет не о выборе цвета галстука и даже не о выборе технологии для корпоративного портала, а о построении такого сложного элемента корпоративной архитектуры, как хранилище, разновидностью которого и является Data Lake, цена ошибки приближается к миллионам долларов. Думаю, это тот самый случай, когда дешевле изучить опыт других, чем приобретать свой.

Именно поэтому я написал эту статью. На самом деле хотелось бы пройтись по вполне реальным примерам из практики последних лет, когда заказчик с радостью бросается на минное поле в надежде, что тут решат все его проблемы.

НО ОБО ВСЕМ ПО ПОРЯДКУ

Как это ни парадоксально, одна из проблем, способная погубить не только Data Lake, но и вообще любое хранилище, сокрыта в самом определении озера данных, которое гласит: «Data Lake — это хранилище данных, содержащее большие объемы сырых данных в их родном формате до тех пор, пока они не понадобятся». Точнее, проблема не в самом определении, а в том, как его интерпретируют, и основной вопрос не в том, какие объемы считать большими, тут все просто — распределенные системы хранения становятся экономически целесообразными, когда речь заходит о сотнях терабайт и петабайтах. Главное недопонимание в другом. Привлекательна сама идея: «Зачем строить хранилище со Staging Area и сложными процедурами ETL? Мы просто возьмем побольше дисков и будем складывать туда данные как есть, а когда они нам понадобятся, мы попросим наших программистов достать их в нужном нам формате. Тем самым мы вроде бы размазываем немалую стоимость проектирования и внедрения хранилища на период его эксплуатации». *И эта идея встречалась у наших заказчиков не единожды.*

«A DATA LAKE IS A STORAGE REPOSITORY THAT HOLDS A VAST AMOUNT OF RAW DATA IN ITS NATIVE FORMAT UNTIL IT IS NEEDED».
TECHTARGET

Ее порочность состоит в одном простом факте: системы источников не идеальны, а если даже и приближаются к идеалу, они не статичны и развиваются. Думаю, каждый, кто хоть раз пытался анализировать данные или организовать их миграцию из legacy-систем, сталкивался с тем, что во вполне **структурированной** базе встречаются данные, которые из этой структуры выбиваются. Где-то новый разработчик написал код, не разобравшись со старыми

таблицами, где-то в новой версии ПО решили поменять структуру хранения и теперь данные лежат в 2, 3, 5 разных таблицах… Особенно это усугубляется тем, что некоторые данные в таких таблицах пересекаются, дублируются и даже противоречат друг другу. В такой ситуации на этап анализа данных даже у закаленных в боях аналитиков уходит масса времени и сил Data Lake без правильной организации метаданных и их структуры умножает эту проблему на порядки. Вы или не сможете доверять результатам, полученным из DL, или будете тратить на элементарные отчеты месяцы. А самое страшное — без правильно организованных метаданных вы, скорее всего, даже не будете знать, какие данные у вас есть, а каких нет. Такое озеро очень быстро покрывается ряской, зарастает травой и превращается в болото данных: все, что в него попало, смогут найти только археологи.

Надо сказать, что немалую толику в нынешней популярности DL сыграло бурное развитие технологий машинного обучения, ведь сейчас всем известно: чем больше накопленных данных, тем лучше можно обучить модель и, значит, получить хороший результат. Даже если у нас данных не хватает, вокруг их полным-полно — это и социальные сети, и мониторинг Wi-Fi-устройств, и сбор кликов, и еще много чего, вплоть до прогноза погоды и фазы Луны. Однако, перефразируя слова из известного фильма, «это не те данные, которые вы ищете». Да, Интернет наполнен историями о том, как простые кофейни повысили маржу, собрав данные о Wi-Fi-подключениях и проанализировав проходимость на основании этих данных. Но проблема в том, что такой анализ можно сделать и без привлечения DL, более того, если вы не можете придумать, как использовать новый для себя источник данных, никакая нейронная сеть не сможет за вас это придумать.

Как показывает практика, самые ценные для моделей данные хранятся в обычных корпоративных системах. Но, даже если вы точно знаете, что подключение, например, к вашим данным профилей клиентов из соцсетей позволит поднять качество скоринговой модели на 2%, это еще не повод строить DL. Простая попытка посчитать экономическую выгоду может привести к удручающему результату: стоимость этих 2% качества может не окупиться и за 15 лет, не так много в России бизнес-структур с подобным горизонтом планирования. В эту ловушку в последнее время, кроме заказчиков из сферы ретейла и страхования, попадают даже банки.

Другая особенность подхода «храним все как есть» в том, что даже при использовании относительно дешевого оборудования позволить его при большом потоке данных могут «не только лишь все». Если вы попытаетесь сохранить поток данных в 10 Гбит/с как есть, то столкнетесь не только с тем, что для этого потребуются поистине астрономические объемы дискового пространства, но и с тем, что для обработки таких объемов нужно больше времени, чем для их накопления.

Не так давно наше законодательство подвигло телеком-операторов массово строить себе хранилища как раз по этому принципу: сырые данные как есть. Однако бизнес не был бы бизнесом, если бы не попытался получить из этих простых хранилищ какую-то выгоду и для себя. И тут мы столкнулись с простым фактом: классическим способом, т.е. на ежедневной основе в режиме batch, мы не можем обработать весь тот объем сырых данных, который накопился за те же сутки. По факту, для того чтобы построить витрины, которые можно было бы использовать в том же процессе машинного обучения или для любой другой аналитики, пришлось применять другой подход — streaming. Данные нужно обрабатывать на лету, укладывая их по мере поступления в требуемую структуру. И получается, что мы вроде бы имеем DL с неструктурированными данными, но все равно вынуждены держать их структурированную копию, для того чтобы ими можно было воспользоваться.

ВАШИ ДАННЫЕ МОГУТ НЕ ИМЕТЬ СТРУКТУРЫ, НО ВАШИ МЕТАДАННЫЕ ОБЯЗАНЫ ЕЕ ИМЕТЬ.

Но, пожалуй, самое главное заблуждение, связанное с DL, — это то, что его внедрение несет Business Value само по себе. Очень показательна история одного банка, который на протяжении полугода рассылал RFI, в которых сначала предлагалось построить аналитику на уже внедренном распределенном хранилище, потом просто построить аналитику, возможно с использованием хранилища, и выдать рекомендации по его использованию, а потом уже RFI на проведение аудита возможности использования хранилища как такового, без какой-либо аналитики. Это классический и, к сожалению, далеко не единственный случай, когда в погоне за модными технологиями и в ожидании магического эффекта от внедрения было построено распределенное хранилище на технологиях Hadoop и BigData, но реальной задачи, для которой бы выбирались технологии, по факту не было.

НЕМАЛУЮ ТОЛИКУ В НЫНЕШНЕЙ ПОПУЛЯРНОСТИ DL СЫГРАЛО БУРНОЕ РАЗВИТИЕ ТЕХНОЛОГИЙ МАШИННОГО ОБУЧЕНИЯ, ВЕДЬ СЕЙЧАС ВСЕМ ИЗВЕСТНО: ЧЕМ БОЛЬШЕ НАКОПЛЕННЫХ ДАННЫХ, ТЕМ ЛУЧШЕ МОЖНО ОБУЧИТЬ МОДЕЛЬ И, ЗНАЧИТ, ПОЛУЧИТЬ ХОРОШИЙ РЕЗУЛЬТАТ.

Характерно, что сами по себе эксперименты подобного плана стоят совсем недешево, ведь для построения простейшей распределенной системы нужно как минимум 5 серверов, причем 2 из них будут заниматься только управлением. Эксперименты с выбором оптимальной архитектуры, поставщика, программного обеспечения могут позволить себе только гиганты рынка или профильные компании, которые строят на этом бизнес. При этом мало построить такое хранилище, оно должно еще правильно эксплуатироваться, и тут нужно понимать, что специалистов в подобных технологиях на рынке крайне немного и, как я уже писал выше, позволить их себе могут не все.

Поэтому, если вдруг вас посетила идея о том, что неплохо бы срочно построить DL, вам нужно честно ответить себе на вопрос: что DL даст мне такого, чего не сможет дать старый добрый DWH? Если причина найдена, строить DL нужно не с выбора технологий и покупки серверов, а с процесса, ради которого вы собираетесь его построить. Как только вы сможете внятно описать процессы и поймете, что затраты на внедрение DL будут перекрыты выгодами, которые вы получите, можно начинать прототипирование.

На самом деле современные технологии замечательны именно тем, что, для того чтобы построить вполне живую и рабочую систему, вам не нужны серверы и дорогие проекты внедрения. Все это понадобится только тогда, когда встанут вопросы надежности и производительности, а на первом этапе зачастую достаточно мощностей обычной рабочей станции. Вы относительно легко сможете масштабировать построенное решение на ЦОД, в крайнем случае вы можете арендовать облачные ресурсы для проведения кратковременных экспериментов.

Не бойтесь пробовать разных поставщиков и ПО, зачастую при внешне одинаковом заявляемом функционале конечное удобство использования может отличаться в разы. Уже на этом этапе вам нужно задуматься о таких вещах, как качество данных, которые вы собираетесь хранить *(Data Quality)*, и управление их метаданными *(Data Governance)*. Ваши данные могут не иметь структуры, но ваши метаданные ОБЯЗАНЫ ее иметь, в противном случае с течением времени вы неизбежно получите то самое болото.

И если все вышеперечисленное — приятные хлопоты для вас, я жду ваше резюме, а если вы не уверены в своих силах, но очень хотите построить хорошее и полезное озеро данных, вам стоит обратиться в компанию, для которой эта работа является профильной.