12 типичных ошибок при бэкапе баз данных

http://habrahabr.ru/post/267881/

Contents

Изначально эта статья задумывалась только для разработчиков и администраторов СУБД Firebird, но после общения с администраторами других БД выяснилось, что большинство ошибок общие, и на очень похожие грабли наступают буквально все. Если Вы можете что-то добавить к этому списку (пусть даже специфическое для конкретной СУБД), пишите в личную почту или в комментариях.

Наша компания занимается инструментами восстановления, резервного копирования, оптимизации и поддержкой СУБД (в основном Firebird, но есть и MSSQL, PostgreSQL, InterBase и др.) и, как результат многочисленных аудитов и ремонтов, накопила коллекцию ошибок, связанных с резервным копированием. Все пункты ниже изложены по мотивам реальных случаев с повреждением баз, потерей и повреждением бэкапов, дисков, сбоями серверов, и прочих «радостей» администраторов БД.

Хотелось бы о них рассказать, чтобы администраторы и разработчики могли поменять свои подходы к управлению бэкапами и предупредить возможные проблемы.

Итак, приступим.

1. Удаление предыдущей копии бэкапа до того, как будет создана новая копия бэкапа

Чаще всего эту ошибку совершают новички, которые не понимают, что основная цель существования резервной копии БД – обеспечить минимальный простой информационной системы (важной частью которой является БД), а не просто создание копии БД.
В результате, с момента удаления последнего бэкапа до создания нового, система находится в незащищенном состоянии, потому что в этот период у базы данных нет ни одной резервной копии. Так как бэкап может создаваться достаточно долго, то это идеальное время для срабатывания закона Мерфи. Особенно хорошо этот подход работает в связке с пунктом 7 (см. ниже).
Рекомендация: не удаляйте предыдущий бэкап до того момента, как создан новый! (и не делайте новый бэкап в уже существующий файл)

2. Перезапись существующей базы данных при восстановлении из бэкапа

Эту ошибку совершают реже, но вот результаты могут быть гораздо печальнее. Если бэкап не проверялся и был поврежден (см. пункт 6), то в результате перезаписи у вас не будет ни предыдущей копии базы, ни валидного бэкапа.
Обычно это безобразие случается в пятницу вечером, в момент дерготни, неразберихи и противоречивых указаний со стороны начальства. Немного отрицательного везения и томные выходные в серверной обеспечены.
У Firebird есть некая защита от этой ошибки – создание рестора из бэкапа с помощью утилиты gbak с дефолтным ключом –create не сработает в случае, если указанное имя файла указывает на существующую БД. К сожалению, есть и обход этой защиты – переключатель –rep, который позволяет-таки переписать существующий файл.
Рекомендации: никогда не перезаписывайте файл боевой БД, не получив письменного указания руководства и, желательно, не получив предложения о новой работе.

3. Использование одношагового бэкапа-рестора, без создания промежуточного файла бэкапа

Стандартные потоки ввода-вывода позволяют провернуть с многими СУБД (с Firebird в том числе) интересный трюк: выполнять потоковый бэкап с немедленным восстановлением БД из него. В результате не создается промежуточный файл бэкапа. Это удобно для проведения регламентных работ и запуска проверочного восстановления (при наличии другой резервной копии), но ни в коем случае не надо использовать это для автоматического бэкапа!
Если в процессе такого бэкап-рестора произойдет серьёзный сбой диска, например, то может повредиться исходная база данных, а новая еще не будет создана. Конечно, если п.1 соблюдается, и есть копия БД от предыдущей попытки, то произойдет только потеря данных, которые были созданы или изменены в БД с момента создания ее копии.
Рекомендации: не используйте одношаговый бэкап-рестор в автоматическом режиме, а в ручном всегда проверяйте наличие достаточно свежей копии.

4. Хранение бэкапа и базы данных на одном и том же физическом устройстве

Тут многие могут посмеяться, что советы мы какие-то детские даем – это же азбука системного администрирования. Так-то оно так, но в связи с распространением виртуальных сред и БД, и диск могут находиться на одной СХД. А она обязательно сломается в самый неподходящий для бизнеса момент. Плюс, все еще существуют люди, которые верят в то, что если они используют RAID (от 1 и выше), то с их данными вообще ничего не может случиться. Еще есть люди, которые верят в сверхнадежность «брендового» железа, но это особый случай.
Рекомендации: не храните бэкап и БД на одном и том же физическом устройстве, каким бы надежным оно не казалась.

5. Отсутствие проверки успешного окончания бэкапа

Вот это довольно частая ошибка и администраторов, и руководителей ИТ подразделений. Если результат бэкапа не проверять, то можно бэкап не делать вообще — результат в общем тот же. Обязательно нужны нотификации по email об успешном бэкапе, а еще лучше и по СМС. Причем, отсутствие нотификации это признак проблемы!
А причем тут руководители, спросит внимательный читатель, дочитавший до этого места? А притом, что администратор обычно бэкап настроит, но вот нотификации ему проверять лень, тем более что лежат они у него в отдельной папочке, и поэтому руководителю ИТ-отдела надо периодически запрашивать дополнительный отчет о состоянии всех бэкапов. Это к вопросу, кого наказывать, если бэкапы вроде есть, но в нужный момент их не оказалось :)
! А при комбинировании с пунктом 2 получаем отсутствие и базы, и бэкапа.
Рекомендации: использовать инструменты автоматизации бэкапов, которые умеют отслеживать успешные и неуспешные бэкапы, сообщать пользователям о проблемах, и имеют обзорные

средства контроля (особенно актуально, когда нужно контролировать десятки и сотни бэкапов на разных серверах).

6. Отсутствие валидации бэкапов

То, что бэкапы куда-то кладутся, не означает, что они оттуда могут быть прочитаны.
Поэтому обязательна периодическая верификация создаваемых бэкапов, чтобы быть уверенным, что создаваемые бэкапы не повреждены, не были скопированы в /dev/null
Рекомендации: никому не доверять, даже себе. Всех надо проверять.

7. Отсутствие health check базы данных при использовании неверифицированных бэкапов

Обычно СУБД имеют несколько видов бэкапа – дампы, просто бэкапы и т.д. Не вдаваясь в конкретику, можно выделить 2 категории – верифицированные и неверифицированные бэкапы. У Firebird это gbak и nbackup.
Gbak читает всю БД на уровне записей для создания файла бэкапа, и создает БД путем вставки записей в новую БД, и таким образом верифицирует и бэкап (есть варианты, как ошибки могут просочиться в отресторенную копию, но это уже другой вид факапа администратора БД, связанный с неверной миграцией), и саму базы данных (если она может быть прочитана от начала до конца, то с большой долей вероятности она не повреждена).
Nbackup (он же инкрементальный бэкап) – временно блокирует основной файл БД на запись (в консистентном состоянии), и позволяет быстро скопировать файл базы данных (полностью или частично/инкрементально).
Для больших БД Firebird (более 500Гб), предпочтительно делать nbackup, чтобы не тормозить работу пользователей, но при этом нужно проверять БД, ведь созданные неверифицированные бэкапы – это страничные копии БД, и если ошибка гнездится на уровне записей (такое случается из-за сбоя RAM) или на логическом уровне, то неверифицированный бэкап будет ее содержать так же, как и оригинальная БД.
Для этого нужно использовать онлайн-валидацию исходной базы данных (в Firebird начиная с версии 2.5.4 доступна онлайн-валидация при помощи gfix, а наш инструмент FBDataGuard поддерживает онлайн-проверку БД для версий 1.5-2.5).
Также, в дополнение к неверифицированному бэкапу желательно периодически (раз в неделю, например) делать верифицированный бэкап.
Для других СУБД необходимо использовать соответствующие средства и комбинации проверок.

8. Отсутствие контроля за свободным местом для бэкапа

В общем-то, это классическая ошибка — при недостатке места бэкап занимает все свободное место и аварийно завершается. При размещении бэкапа на одном диске вместе с БД может привести к остановке работы с БД, при размещении на системном диске – к поломке системы.
В комбинации с пунктом 4 в лучшем случае получим остановку работы системы, потому что базе данных тоже нужно свободное место, а оно кончилось из-за бэкапа.
А в комбинации с пунктами 5 и 2 опять получаем в результате отсутствие и базы, и бэкапа.
Рекомендации: использовать инструменты для бэкапа, которые делают прогноз размера бэкапа и предупреждают о возможной нехватке места.

9. Отсутствие контроля времени продолжительности бэкапа

Буквально полгода назад бэкап длился 40 минут, и вдруг стал 3 часа – почему? Возможно, вырос размер БД, а может быть, выпал диск из RAID-массива, отчего скорость записи резко деградировала, и все ваши бэкапы вот-вот покинут этот бренный мир. А может быть, добрый коллега одновременно включил еще одну систему резервного копирования (кстати, в Firebird

можно запустить несколько бэкапов сразу, только не очень понятно, зачем это может быть нужно).
Если не контролировать время исполнения бэкапа, то можно проглядеть возникшую проблему и упустить шанс исправить ее до того, как она станет большой.
Также, в случае, если система резервного копирования не отслеживает состояния заданий, а запускает их просто по графику, легко можно попасть на «утренний троллейбус» — это когда новый бэкап может начаться в момент, пока предыдущий еще не закончился.
Рекомендации: использовать средства контроля продолжительности процесса бэкапа.

10. Исполнение бэкапа БД во время применения апдейтов ОС

Очень частая проблема, особенно в комбинации с п.9 и включенными автоматическими апдейтами Windows (которые по умолчанию происходят в 3 ночи). В лучшем случае приводит к замедлению процесса, а если ОС перегружается для применения апдейтов, то бэкап будет испорчен. Хорошо еще, что апдейты ОС не каждый день случаются.
Рекомендации: Если нельзя отключить, то назначьте апдейты ОС на такое время, когда они не смогут помешать бэкапам.

11. Бэкап базы данных средствами файлового бэкапа или средствами бэкапа виртуальной машины целиком, при включенном сервере БД

Многие администраторы забывают о том факте, что любая СУБД имеет активный и сложный кэш, в котором содержатся читаемые и записываемые данные, а сами файлы БД открываются в режиме случайного доступа.
Именно поэтому необходимо использовать специальные виды бэкапа, а не простой файловый бэкап (включая простое копирование файлов БД) или бэкап виртуальной машины. Файловые средства бэкапа читают БД последовательно и, особенно для больших БД, могут идти продолжительное время, поэтому нельзя гарантировать целостность созданной копии.
Для желающих осуществлять бэкап БД с помощью файловых средств или средств бэкапа виртуальных машин можно предложить 2 способа:

1. полностью выключать сервисы и процессы СУБД, чтобы ничего не находилось в кэше,
2. использовать агенты и/или скрипты, переводящие базы данных в специальный режим, когда копирование файла БД последовательным образом является безопасным.

Для Firebird необходимо блокировать основной файл БД с помощью nbackup до начала резервного копирования и разблокировать после окончания, для других СУБД есть аналогичные средства включения/выключение соответствующих режимов.

Кое-кто из администраторов БД убежден, что если в СУБД есть лог транзакций, то такую БД можно безопасно бэкапить стандартными файловыми средствами, в крайнем случае будет повреждение только этого лога. Это опасное заблуждение, которое не поддерживается производителями СУБД.

Корни этого заблуждения понятны: агрессивная реклама от производителей виртуальных машин и от производителей средств бэкапа обычно умалчивает о том, что для БД и других активно изменяемых файлов необходимы дополнительные настройки. Не доверяйте рекламе — не все йогурты одинаково полезны.
Рекомендации: не используйте файловые средства бэкапа и средства бэкапа VM без соответствующих средств автоматизации.

12. Подмена бэкапа репликацией

Бэкап и репликация данных служат повышению надежности и предотвращению потери данных, но все же существенно отличаются.
Все любят репликацию за способность синхронизировать данные на другом сервере с минимальном задержкой, однако и у бэкапа есть ряд неоспоримых преимуществ. Например, в случае случайного (или намеренного) удаления данных репликация быстро и невозмутимо оттранслирует изменения на реплику, а бэкап, особенно на read-only носителе, к таким операциям невосприимчив. Настроить правильную репликацию, как и правильный бэкап, стоит определенных усилий, и вероятность ошибок там также существует.
Рекомендации: Если у вас настроена репликация, не пренебрегайте бэкапами, используйте их совместно.

Выводы

Правильно организовать резервное копирование для вашей любимой СУБД не так-то и просто, поэтому обычно администраторы БД из организаций, где ценят данные, обычно используют профессиональные инструменты для бэкапов, которые позволяют учесть и предотвратить описанные выше ошибки. Для Firebird (уж простите за рекламу) есть наш FBDataGuard, для других СУБД можно использовать DBArtisan или другие средства.

Ну, и конечно – не забывайте холить и лелеять свою админскую паранойю, например, возьмите и проверьте свои бэкапы… вот прямо сейчас!