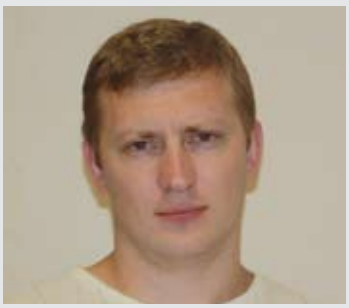
Методы обеспечения надежности работы СУБД

Александр АТЦИК,
руководитель инженерно-аналитического отдела,
НТЦ «Аргус»

Дмитрий СМИРНОВ,
руководитель отдела миграции и поддержки баз данных,
НТЦ «Аргус»



Александр ПЕРЕПЕЛИЦА,
ведущий инженер-программист,
НТЦ «Аргус»

Информационная система на предприятии – это, прежде всего, хранилище данных и информации, являющихся основной ценностью среди всех компонентов системы. Более того, накопленная информация является, как правило, конфиденциальной. Вполне очевидно, что обрабатывать, хранить и структурировать большие объемы информации без набора удобных, функциональных и надежных инструментов достаточно сложно. Такими инструментами в наши дни стали системы управления базами данных (СУБД).

Использование СУБД оказалось чрезвычайно практичным методом работы с большими объемами данных. К наиболее важным задачам СУБД относятся организация быстрого доступа к данным и восстановление данных после сбоев. Вторая задача СУБД становится особенно важной, когда на основании обрабатываемой и хранимой информации строятся все бизнес-процессы, как, например, на информации технического учета – эксплуатационная деятельность оператора связи. Потеря актуальности хранимых данных о ресурсах, услугах или клиентах может вызвать ошибки в процессах продаж или нарушение договоров о качестве обслуживания, что влечет финансовые потери. Именно поэтому ключевыми задачами являются обеспечение бесперебойного доступа ко всем ресурсам и уменьшение потерь данных, возникающих вследствие сбоев.

Обычно на характеристики надежности функционирования баз данных заказчик накладывает строгие ограничения, поэтому разработчики СУБД не должны «скупиться» на различные программные возможности, наилучшим образом решающие задачи работы с данными. Однако с повышением эффективности увеличивается и стоимость подобных приложений, что в некоторых случаях заставляет поставщиков допускать определенные риски во избежание необоснованных издержек. В связи с этим качеству и возможностям СУБД и репутации компании-разработчика систем управления базами данных уделяется особое внимание.

В рамках данной статьи мы рассмотрим основные вопросы и подходы к их решению и постараемся показать, что при выборе методов резервирования данных действуют противоречивые критерии, не позволяющие отдать предпочтение тому или иному подходу и заставляющие рассматривать каждый случай индивидуально.

Угрозы для СУБД

Система СУБД, представляет собой программно-аппаратный комплекс, предназначенный для хранения, организации доступа, управления и восстановления данных. Соответственно, потенциальные угрозы потери, искажения или недоступности информации могут иметь физическую или программную причину.

Обычно в процессе эксплуатации СУБД предусматривается возможность возникновения аварийных ситуаций, которые должны

учитываться на уровне технологий СУБД, и эти решения должны обеспечивать системе требуемый уровень безопасности и надежности. Рассмотрим самые распространенные ситуации.

Вышел из строя один или несколько жестких дисков в хранилище данных – полная или частичная потеря данных. Необходимо иметь возможность произвести восстановление данных, хранившихся на этом жестком диске.

Вышла из строя материнская плата на сервере – сервер, в котором она установлена, станет недоступным, и все недавние изменения или результаты текущей работы сервера будут утрачены. Необходимо иметь возможность обслужить поступающую нагрузку с заданным качеством, которая изначально предназначалась для данного сервера. При этом нужно восстановить данные, измененные в результате завершенных на данном сервере транзакций.

Обрыв связи между сервером и хранилищем – потеря всех текущих изменений, обрабатываемых на данном сервере.

Вирус на сервере – могут возникнуть сбои доступа к серверу, что приведет к потере недавних изменений и остановит работу. Под угрозой также находятся целостность данных в хранилище, архивные логи, потеря которых исключит возможность восстановления утраченной части базы данных. Поскольку поведение вирусов непредсказуемо, последствия тоже могут иметь случайный характер.

Обеспечение отказоустойчивости

В соответствии с концепцией одного из ведущих поставщиков СУБД, используемые инструменты для защиты данных должны обладать следующими свойствами:

- надежность (reliability) – вся информация, необходимая для восстановления данных, должна храниться в надежном месте на надежных носителях и быть зарезервирована;
- гибкость (flexibility) – резервирование должно быть произведено так, чтобы при необходимости можно было восстановить не только всю базу в целом, но и, к примеру, отдельное табличное пространство или конкретные файлы данных;
- управляемость (manageability) – резервные файлы должны быть легко и удобно управляемы для того, чтобы восстановление можно было произвести в кратчайшие сроки;
- готовность (availability) – работа по резервированию ни при каких условиях не должна мешать работе с базами данных и обработке транзакций в БД. Кроме того, работа по восстановлению должна выполняться незаметно для пользователя.

> Физическое разнесение экземпляров БД дает возможность избежать аппаратного сбоя и повышает устойчивость системы к авариям.

Практически все существующие подходы к обеспечению отказоустойчивости систем СУБД можно отнести к описанным ниже технологически нейтральным группам, а конечную реализацию от производителя можно оценивать по свойствам надежности, гибкости и т. д.

Технологически нейтральные подходы к обеспечению отказоустойчивости

Холодное (offline) резервирование. Наиболее простой способ резервирования БД: всю поступающую нагрузку обслуживает основной экземпляр, с которого с определенной периодичностью снимаются точки восстановления путем полного копирования базы данных на носитель. При возникновении сбоя последняя (или более поздняя) точка восстановления загружается на БД (это может быть основная или резервная БД), и на нее переключается вся поступающая нагрузка.

В данном случае возможны три схемы:

- холодное резервирование производится на отдельный носитель этого же сервера;
- холодное резервирование производится на носитель отдельного сервера, предназначенного для резервирования информации компании по сети (скорее всего, в качестве сервера БД он не может быть использован из-за ограниченных ресурсов);
- холодное резервирование производится на отдельный сервер, который выделен в качестве запасного/резервного сервера БД.

Снятие точек восстановления при холодном резервировании можно осуществлять лишь периодически, поскольку при этом подразумевается остановка основной БД.

Периодическое снятие точек восстановления выполняется средствами базы данных или внешними утилитами, копирующими файлы БД на уровне операционной системы.

В случае с внешними утилитами (не интегрированными с БД) временно блокируется доступ к базам данных и происходит копирование управляющих файлов (control file), табличных пространств (tablespaces), а также имеющихся логов (log files) и архивных логов (archive log files) на носитель информации (жесткий диск, магнитная лента). Таким образом, по завершении описанной операции получается точная копия имеющейся базы данных на момент остановки работы с ней.

Важно отметить, что многие поставщики СУБД позволяют осуществлять холодное резервирование без необходимости приобретения лицензий на резервную БД, что существенно снижает стоимость подобных решений.

Неудобство применения данного подхода заключается в том, что частота копирования БД ограничена и, как правило, совпадает с часами наименьшей нагрузки БД. Очевидно, что работы по резервированию

приходится производить либо в ночное время, либо в выходные дни, что, в свою очередь, позволяет производить откаты только к этим моментам времени с возможной потерей информации за весь последний день или неделю. Холодное резервирование находит свое применение с базами данных небольшой емкости. Процесс восстановления данных занимает длительное время и является обратным копированием имеющейся информации с резервного носителя на основной. А если учесть, что продолжительность процесса восстановления – как работоспособности, так и работы в нормальном режиме с использованием основных ресурсов – жестко нормируется с заказчиком, то могут возникнуть ситуации, когда такой подход неприемлем.

Непрерывное снятие точек восстановление. Подход непрерывного снятия точек восстановления заключается в том, чтобы отслеживать изменения, происходящие в БД, в режиме реального времени и сохранять историю изменений на выделенном накопителе. Предполагается, что таким образом любую БД можно «накатить» на любую, заданную администратором дату. Однако такой подход порождает немало проблем, которые приходится решать производителям. К примеру, при частом обновлении данных объем знаний об изменениях в БД может превышать объем исходной БД в несколько раз. Положительным моментом является то, что при сбое теряются только данные незавершенных транзакций, существовавших в момент сбоя. В качестве компромисса может использоваться вариант с периодическим снятием образов БД по расписанию и сохранение непрерывной истории изменений за какой-либо ограниченный промежуток времени (например, за неделю).

Горячее (online) резервирование. Это целый класс способов проведения резервирования баз данных, отличительной чертой которых является то, что для выполнения резервирования нет необходимости останавливать продуктивную БД. В нормальном режиме все запросы обслуживаются основной БД, а изменения в ней синхронизируются с резервной. Резервная база чаще всего недоступна для работы пользователей или же доступна только для чтения. При возникновении сбоя в основной БД вся нагрузка переключается на резервный экземпляр.

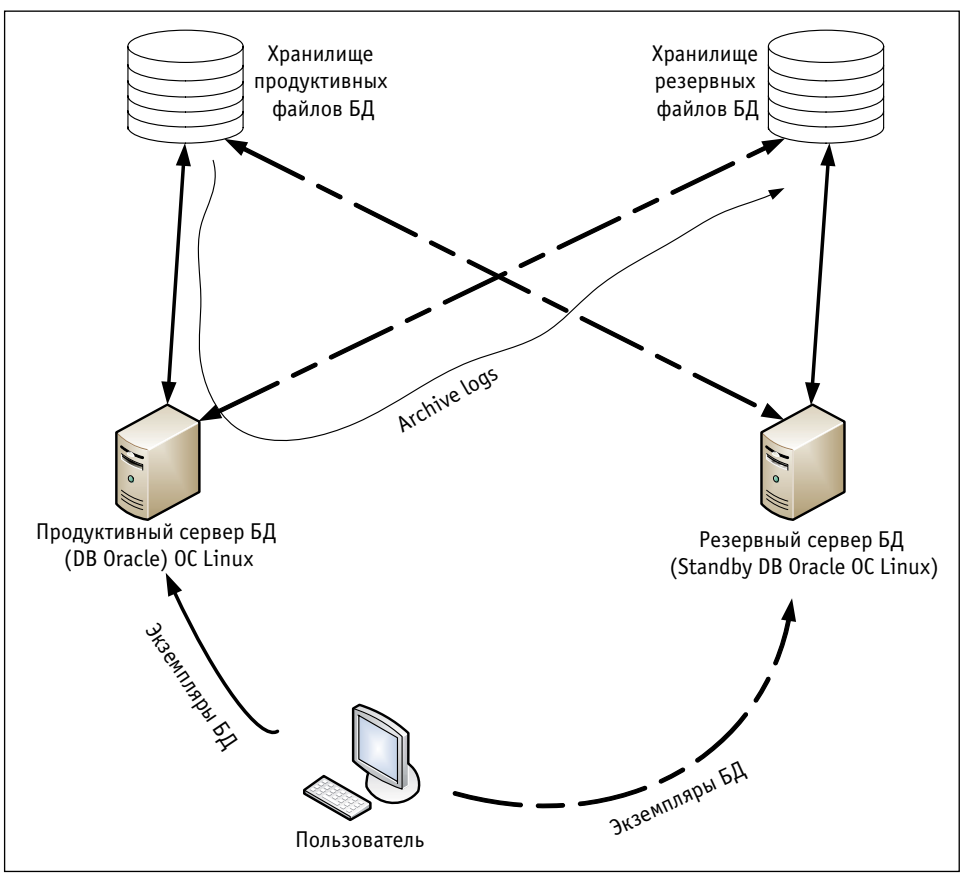
Можно устанавливать любое значение периода снятия точек восстановления, однако, поскольку снятие точки восстановления требует копирования лога изменений, архивации, передачи его в резервную базу и прочих действий, производительность основной базы во время этого заметно падает. Следовательно, необходимо подбирать период снятия точек так, чтобы достигнуть оптимума между падением производительности и уменьшением потери данных при сбое. Часто период снятия лога регулируется через размер лог файла, например, при достижении им 100 Мб происходит переключение на следующий лог, а завершенный отправляется в резервную базу. Чтобы перейти от размера файла к периоду снятия точки восстановления, обратимся к одному из наших решений, в котором БД используется для хранения данных централизованного бюро ремонта на 100 пользователей: в течение рабочей смены пользователей переключение лог файлов размером 100 Мб происходит приблизительно раз в два часа.

Физическое разнесение экземпляров БД дает возможность избежать аппаратного сбоя и повышает устойчивость системы к авариям. Такой подход позволяет существенно повысить надежность хранения данных при возникновении сбоев в одном из экземпляров БД. В простейшем способе организации данного подхода требуется 100%-ная избыточность хранения данных, а значит, существенно увеличивается стоимость данного решения.

В случае осуществления горячего резервирования для резервной базы (Stand-by) большинство ведущих поставщиков СУБД требуют полноценного лицензирования, аналогичного лицензированию продуктивной базы.

Стоимость решения можно снизить за счет использования в качестве резервного менее мощного сервера с меньшим количеством процессоров (лицензирование для СУБД производится на основе количества пользователей или процессоров в сервере БД), а в прикладном ПО

Схема резервирования в одном из OSS-решений



предусмотреть возможность блокировать некоторую группу пользователей до тех пор, пока не будет восстановлен основной сервер БД.

В современных системах нередко используется подход с распределением нагрузки: создаются несколько экземпляров БД, каждая из которых обрабатывает часть запросов. При выходе одного экземпляра БД из строя нагрузка распределяется между остальными. Распределение нагрузки позволяет снизить избыточность хранения.

Одним из вариантов распределения нагрузки является архитектура БД с полной репликацией. В данном случае существуют два экземпляра БД в разных географических регионах, они обслуживают локальную нагрузку. Эти экземпляры БД связываются между собой каналом связи, за счет которого происходит их синхронизация. Такой подход позволяет гарантировать надежность хранения информации (каждая запись сохраняется как минимум в двух экземплярах БД) и экономить на канале связи, поскольку трафик сигнализации будет всегда меньше, чем трафик, создаваемый объемом запросов и ответов БД. Такой способ хорошо подходит в тех случаях, когда количество обращений к БД превышает количество обновлений в них. Тогда процессы синхронизации не требуют значительных ресурсов.

Опыт работы с базами данных и реально применяемые методы обеспечения отказоустойчивости

Большинство из заказчиков предъявляют жесткие требования по обеспечению надежности работы с базами данных, так как в них хранится информация подсистем инвентаризации ресурсов, услуг, клиентов. Потеря актуальности данных может существенно отразиться на бизнесе. Поэтому требования к БД включают в себя нормы времени восстановления системы после аварии, потери информации при возникновении внештатных ситуаций и пр.

В качестве реального примера использования системы резервирования рассмотрим архитектуру БД некоторых из наших заказчиков. Пользователи работают с экземплярами баз данных, обращаясь к основному серверу СУБД, который управляет файлами, хранящимися в основном хранилище. В основном хранилище также располагаются архивированные файлы системного журнала (archive logs). К основному хранилищу подключен резервный сервер в режиме горячего резерва. Он управляет хранилищем резервных файлов БД. Резервная база данных создается с помощью специальной копии управляющего файла, который генерируется на основной базе и на основе копии продуктивных файлов БД, изменяющихся путем чтения архивных логов продуктивной базы данных.

Таким образом, при аварии основного сервера пользователи переключаются на резервный. При этом разница в работе для них не будет ощутима, поскольку последний будет использовать ресурсы основного хранилища. Авария в основном хранилище не повлияет на работу эксплуатационных процессов, поскольку в резерве находится точная копия основного. Вероятность того, что произойдут сбои на обоих серверах или в обоих хранилищах, очень мала. Схематически это показано на рисунке.

К сожалению, в рамках обзорной статьи невозможно охватить все возможные проблемы, влияющие на надежность БД. Удалось описать только основные методы повышения характеристик надежности. Выделить универсальный подход, который был бы дешевым и надежным, невозможно. Максимальная надежность достигается в случае 100%-ного резервирования с непрерывным снятием точек восстановления, а минимальная стоимость – при периодическом снятии образа БД с остановкой основного сервера.

Многие современные разработки баз данных направлены на поиски компромиссных решений, позволяющих уменьшить процент резервирования баз данных и повысить коэффициент цена/надежность таких решений. В конечном итоге схема обеспечения надежности БД определяется индивидуально, исходя из требований заказчика и данных, которые хранятся в базе. А ответ на вопрос «все ли проблемы решены?», возможно, никогда не станет положительным, так как вместе с возможностями БД растут и потребности. ■

мнение специалиста

Александр ХРАМЦОВ,
директор технического департамента
компании «Verysell Проекты»

Действительно, не существует универсального метода резервирования данных, ведь, как писал Л.Н. Толстой: «каждая несчастливая семья несчастлива по-своему»... Задачу резервирования данных и обеспечения надежности СУБД неверно решать в отрыве от качества сервисов, предоставляемых информационной системой в целом. В конечном итоге, данные и предоставление доступа к ним – тоже сервис. Вообще говоря, для решения задачи обеспечения целостности и сохранности данных требуются не только технические знания и ресурсы, но и целый ряд организационных и управленческих мероприятий.

Во-первых, необходимо четко сформулировать требования к механизмам резервирования и/или репликации данных, то есть определить уровни обслуживания (SLA) и параметры восстановления (RPO/RTO) для информационной системы адекватные требованиям бизнеса. Иначе можно получить нагромождение дорогих и высокотехнологичных решений, которые невозможно будет использовать из-за высокой сложности администрирования и несовместимости.

Второй задачей, которой нужно уделить особое внимание является выбор наиболее подходящих методов и механизмов резервирования данных. Делать выбор лучше всего, изучая технологические публикации производителей оборудования и ПО для обработки данных, серьезные аналитические статьи независимых разработчиков и интеграторов, опыт других крупных организаций в этой области.

Наконец, третьей серьезной задачей является разработка регламентов и документов, обеспечивающих восстановление данных и продолжение бизнеса в аварийных условиях. Нужно понимать, чем яснее прописаны обязанности и функции каждого дежурного администратора и пользователя во всех мыслимых ситуациях, тем быстрее информационная система вернется к своему функционалу в полном объеме.