ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ИСПОЛЬЗОВАНИЯ БАЗ ДАННЫХ В УСЛОВИЯХ ЧРЕЗВЫЧАЙНЫХ СИТУАЦИЙ

Александр Кузёмин, Алексей Василенко

Аннотация: *Рассматриваются проблемы уменьшения риска, увеличения надежности и безопасности хранения и использования баз данных (БД) в условиях чрезвычайных ситуаций (ЧС). Предлагается технологическая последовательность решения поставленной задачи. Для анализа данных учитывается особенность представления количественных и качественных параметров в условиях ЧС. Показана возможность логического и функционального ER моделирования CASE технологии с учетом минимизации риска жизнедеятельности в зоне ЧС. Приведено техническое решение поставленной задачи.*

Ключевые слова: *Чрезвычайные ситуации, База Данных (БД), архив, резервирование, система управления базой данных (СУБД), Сервер.*

Постановка задачи исследования

На сегодняшний день информационные системы (общегосударственные, региональные, ведомственные и др. ситуационные центры) в условиях опасности возникновения чрезвычайных ситуаций (ЧС) играют важную роль при обеспечении безопасности жизнедеятельности населения [Kuzomin & Torojev, 2006]. При возникновении риска возникновения ЧС и при ликвидации последствий ЧС необходимо сохранить, обработать и использовать жизненно важную информацию для принятия управленческих решений. На сегодняшний день информационные системы (общегосударственные, региональные, ведомственные и др. ситуационные центры) в условиях опасности возникновения ЧС играют важную роль при обеспечении безопасности жизнедеятельности населения. При возникновении риска возникновения ЧС и при ликвидации последствий ЧС необходимо сохранить, обработать и использовать жизненно важную информацию для принятия управленческих решений. Однако ЧС порождают и риски потери информации – данных, которые определяют надежность и живучесть баз данных в условиях ЧС.

Повышение риска уменьшения надежности и безопасности хранения и использования БД связано с возникновением в условиях ЧС аппаратных и программных сбоев. Риски уменьшения надежности баз данных (БД) определяются соответствующими источниками отказа (Рис. 1). В условиях ЧС необходимо обеспечить живучесть ситуационного центра и в частности БД. Под живучестью БД следует понимать свойство БД выполнять своё функциональное назначение с минимальным риском и максимальной надежностью в условиях ЧС.

Целью данной статьи является представление результатов разработки информационной технологии, которая направлена на уменьшения риска отказа БД, повышение надежности и живучести баз данных ситуационных центров в условиях ЧС.

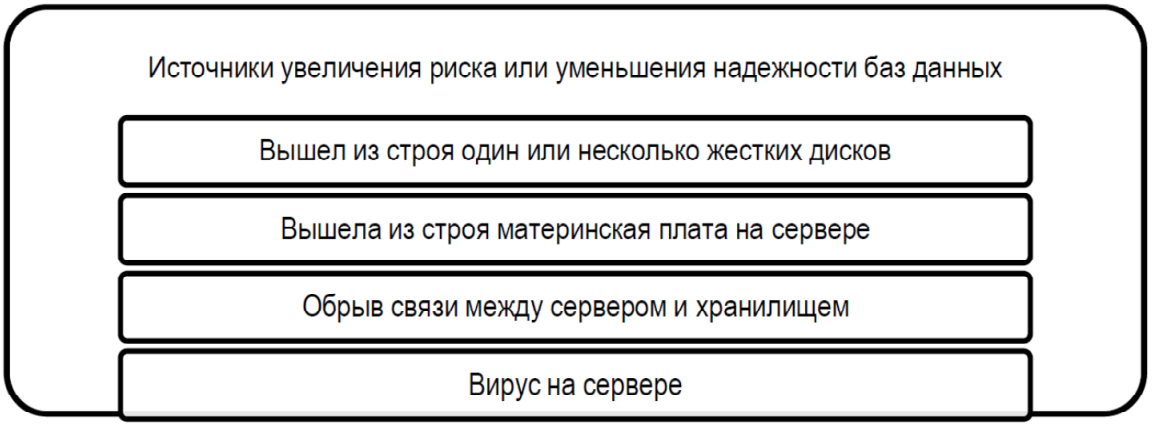

Рисунок 1. Источники увеличения риска или уменьшения надежности баз данных

При этом можно определить следующую технологическую последовательность решения поставленной задачи:

1. Техническое задание на разработку БД.
2. Анализ качества данных для ситуационного центра.
3. Выбор архитектуры программно-аппаратного комплекса ситуационного центра [1].
4. Разработка модели управления БД (СУБД).
5. Оценка рисков и надежности БД в условиях ЧС.
6. Возврат к пункту 3, если допустимые значения оцениваемых критериев превышаются.
7. Минимизировать коэффициент цена/риск или цена/надежность.
8. Возврат к пункту 3, если допустимые значения оцениваемых критериев превышаются.
9. Окончательный выбор архитектуры в соответствиями с требованиями ТЗ.

Рассмотрим ключевые этапы предлагаемой технологии.

Анализ качества данных для ситуационного центра.

Для проектирования ситуационного центра необходимо выполнить системный анализ данных, который в первую определяет проектные решения для БД. При проектировании баз данных доминирующее значение приобретают сами данные, их хранение и обработка. Однако их содержание для применения к качеству баз данных требуется уточнить. Выделяемые показатели качества должны иметь практический интерес для пользователей и быть упорядочены в соответствии с приоритетами практического применения. Для оценивания качества информации, которая должна использоваться в ситуационном центре может применяться общий подход к выделению адекватной номенклатуры стандартизированных в ISO 9126 базовых характеристик и субхарактеристик [Липаев, 2001]. Каждый выделяемый показатель качества должен быть пригоден для достоверного экспертного оценивания или измерения, а также для сравнения с требуемым значением.

Мерой качества функциональной пригодности БД (Рис. 2) может быть степень покрытия целей ситуационного центра, назначения и функций в доступной пользователям форме.

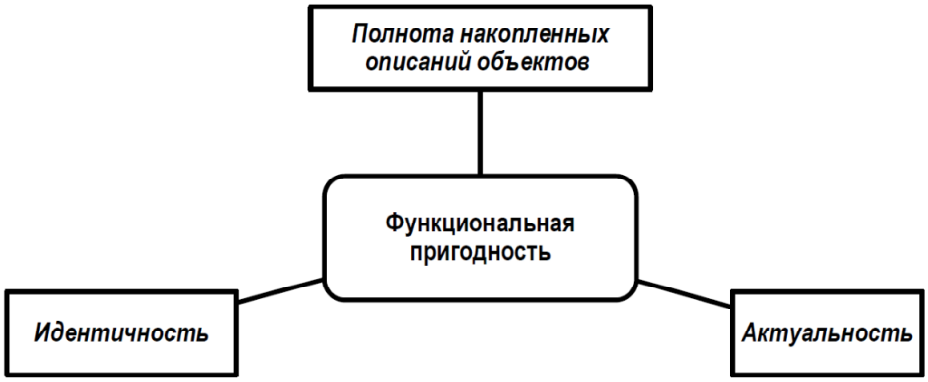

Рисунок 2. Проблема функциональной пригодности

Корректность или достоверность данных — это степень соответствия данных об объектах в базах данных реальным объектам в зоне возникновения ЧС, в данный момент времени, определяющаяся изменениями самих объектов, некорректностями записей об их состоянии или некорректностями расчетов их характеристик. Сюда же можно отнести и некоторые объемно-временные характеристики сохраняемых и обрабатываемых данных (Рис. 3).

Рисунок 3. Объемно-временные характеристики БД

Защищенность информации реализуется средствами СУБД в сочетании с поддерживающими их средствами защиты данных. Цели, назначение и функции защиты тесно связаны с особенностями функциональной пригодности каждой базы данных. В распределенных БД показатели защищенности тесно связаны с характеристиками целостности и отражают степень тождественности одинаковых данных в памяти удаленных компонентов ситуационного центра.

Кроме окружающей среды, в ситуационном центре учитывается ресурс. Этот ресурс определяется средой технических и организационных средств, который может использовать лицо принимающее решение (ЛПР) для предупреждения или ликвидации последствий ЧС. Среди отмеченных выше данных имеются количественные и качественные параметры. Количественные параметры в БД традиционно представляются в виде таблиц. Для использования БД в ситуационном центре выполняется статистический анализ данных, который позволяет выделить микроситуации, которые соответствуют наиболее влияющим параметрам окружающей среды в зоне действия ЧС [Kuzemin & Lyashenko, 2007; Kuzemin et al, 2007]. Микроситуации для прошедших ЧС могут быть использованы для наполнения прецедентной базы количественных данных, для которых выделены наилучшие решения в виде качественных параметров. Качественные параметры представляют любое решение или действие ЛПР, которое имеет три основные характеристики: цель действия; описание действия; средство его выполнения.

Все эти характеристики можно получить из ответов специалистов и экспертов, которые могут квалифицированно ответить на следующие три вопроса: "Какой должен быть результат?" – ответ будет характеризовать цель (назначение действия); "Что делать?" - ответ будет давать описание действия; "Как делать?" - ответ будет характеризовать средство выполнения (умение и возможность). Ставятся три вопроса, на которые необходимо ответить эксперту или ЛПР – «КТО», «ЧТО» и «КАК».

Для предлагаемой технологии разработки ситуационного центра для подсистемы ликвидации лавинной опасности рассмотрим пример «Ликвидация последствий схода лавины». В результате анализа данных (регрессионного, диссперсионного и кластерного анализа) были выявлены наиболее влияющие параметры (3 – 5), которые определялись как микроситуаци с наибольшей вероятностью (0.9 – 0.95) приводили к удачным или рациональным решениям при возникновении или ликвидации последствий ЧС [Куземин & Сорочан, 2004]. При возникновении ЧС принимались соответствующие управляющие решения, которые можно было бы оценивать по своей эффективности для обеспечения наименьших рисков жизнедеятельности, социальных и материальных потель. Такие удачные решения составляли основу прецедентной БЗ и используются при поиске управляющих решений с минимальными рисками и минимальным временем на выполнение. Для прецедентной БЗ были установлены соответствующие понятия, категории, отношения и сценарии действий для предупреждения и ликвидации последствий ЧС [Диго, 2005].

В процессе фиксации ситуации и ликвидации последствий ЧС для рассматриваемого примера на вопрос «КТО» можно выделить таких «Исполнителей»: гражданин, заметивший возникновение ЧС, сотрудники снеголавинной службы министерства по чрезвычайным ситуациям (МЧС), диспечер, ответственное лицо службы МЧС, эксперты МЧС. На вопрос «ЧТО» могут использоваться «Документы» для описания действий: метеорологическая сводка, заявка о проишествии, лавиноопасная экспертиза, рекомендации по проведению протилавинных мероприятий, акт о последствиях.

«Действия участников при ликвидации последствий ЧС» могут носить предметный характер: *сообщение, фиксация, ликвидация, регистрация последствий.*

Проблемная, контролируемая «Микроситуация на языке представления ситуаций» выглядит следующим образом:

- микроситуация с порядковым номером;
- гражданин замечает физические изменения в снежном прокрове;
- сотрудники снеголавинной станции объявляют штормовое предупреждение;
- сотрудники снеголавинной станции сообщают диспечеру службы МЧС;
- диспечер выясняет место расположение возможного схода лавины;
- диспечер передает первоначально обработанную информацию совету экспертов;
- эксперты производят анализ и принимают решения;
- сотрудники служба МЧС реализуют выработаные рекомендации относительно лавинной опасности;
- эксперт службы МЧС оценивает последствия проишествия.

Понятия, выявленные для данной ситуации распределяются в соответствующих иерархиях категорий таким образом, что бы можно было отобразить дерево аналалогичных функциональных структур для ликвидации последствий ЧС.

Подсистема службы МЧС для ликвидации лавинной опасности на первом этапе принятия решений может быть определена как регистрационная структура системы управления, которая по функциональному назначению аналогична соответствующей структуре МЧС в пожарной части.

Категория «Документы» отражает в документах действия, которые требуются при описании управленческих действий для ликвидации лавинной опасности, в иерархии категории. «Метеорологическая сводка» соответствует перечню документов, который соответствует подключению всех необходимых служб обеспечения безопасности жизнедеятельности. «Лавинная экспертиза» необходима для отображения причин схода лавины и установления возможных противолавинных сооружений. Акт о последствиях помогает определить величину материального ущерба, вызванного сходом лавины.

Сотрудники МЧС и эксперты МЧС выполняют определенные функции – анализ поступивших данных, принятие решений и выработку рекомендаций по проведению противолавинных мероприятий, ликвидации лавинной опасности и проведению экспертного анализа ситуации (выявление причин и последствий).

Штормовое предупреждение и проведение лавиноопасной экспертизы являются контролирующими действиями из категории «Управленческие действия».

Обозначим понятия следующим образом: c_1 – гражданин, c_2 – диспечер, c_3 – команда МЧС, r_1 – сообщить, r_2 – проверить, r_3 – ликвидировать, e_1 – снеголавинная служба, e_2 – сообщения очевидцев, e_3 – штормовое предупреждение, e_4 – экспертиза, e_5 – справка о последствиях, e_6 – база данных.

Тогда контролируемая или проблемная микроситуация $MSit_1$ будет выглядеть так:

$$MSit_1 = \{c_1 r_2 e_5, c_1 r_2 e_4, c_3 r_1 e_1, c_2 r_2 e_2, c_2 r_2 e_3, c_2 r_2 e_6, c_2 r_3 e_7\}. \quad (1)$$

После формирования описания ситуации выполняется запрос к прецедентной БЗ на поиск близких микроситуаций [Куземин & Сорочан, 2004]. Была найдена микроситуация, описывающая вызов пожарной команды. Для этой микроситуации была дана самая высокая оценка по показателю качества принятого

решения – минимуму рисков социального и экономического для обеспечения безопасности жизнедеятельности людей и инфраструктуры в районе ЧС. Была получена следующая микроситуация с описанием действий или решений:

- микроситуация с порядковым номером;
- клиент вызывает пожарную команду;
- диспечер фиксирует заказ;
- диспечер передает заказ свободной машине;
- команда производит выезд и ликвидацию пожараы.

В символьном виде данная прецедентная микроситуация $MSit_1'$ представляется так:

$$MSit_1' = \{c_1'r_1'e_1', c_2'r_2'e_2', c_2'r_2'e_3', c_2'r_3'e_4'\}, \quad (2)$$

где c_1' –клиент; c_2' – диспечер; e_1' – пожарная часть; e_2' – вызов; e_3' – приезд пожарной команды на место происшествия; e_4' – ликвидация пожара; r_1' – вызвать; r_2' – фиксировать; r_3' – выполнить.

Для выделенных понятий определялась их «близость» в соответствии метрикой расстояния между понятиями.

Полученные результаты могут быть использованы для созданя базы знаний повторного использования в ситуационном центре при поиске рекомендаций ЛПР как по предупреждению так и при ликвидации последствий ЧС.

Рекомендации для принятия управленческих решений по ликвидации лавинной опасности, получаются после модификации близкого удачного управленческого решения для микроситуации вызова пожарной команды. Близость микроситуаций определяется в такой последовательности:

1. Для количественных данных прецедентов множество микроситуаций, полученных в результате статистическго анализа [Kuzemin & Lyashenko, 2007; Kuzemin et al, 2007] на выбраной метрики эвклидового расстояния

$$d_{ik} = \sqrt{\sum_{j=1}^{N} \left(x_{ij} - x_{kj}\right)^2}, d_{ik} \le \Delta,$$

где $d_{ik} \le \Delta$ - условие для оценивания близости микроситуаций, x_{ij}, x_{kj} - количественные данные сравниваемых микроситуаций.

2. Для качественных данных в соответствии с метрикой сходства Хэмминга определяется схожесть процессов, т.е. ответов на вопрос «ЧТО». Похожесть процессов и действий в первую очередь заключается в фиксации вызова, связанного с возникшей ЧС. Действия еще могут быть такими: проверка и обработка полученной информации о ситуации, принятие решения и реализация рекомендаций по ликвидации. Функциональный блок «Зафиксировать вызов о пожаре» заменен на «Зафиксировать штормовое предупреждение».
3. Для ЛПР выдается рекомендации по предупреждению или ликвидации последствий ЧС выбпраных управленческий решений – ответы на вопросы «КТО» и «КАК».

Выбор архитектуры программно-аппаратного комплекса ситуационного центра

В рамках данной статьи рассматриваются проблемы проектирования БД и СУБД. БД для ситуационного центра проходят разные этапы своего жизненного цикла, начиная от замысла системы, предпроектного обследования, включая этапы проектирования, эксплуатации, а далее - модернизации системы.

Создание БД для ситуационного центра практически невозможно без использования средств автоматизации проектирования (CASE-систем) [Диго, 2005]. Их использование позволяет не только ускорить работы и повысить качество их выполнения, но и дает инструменты для организации коллективного труда группы проектировщиков. Использование инструментальных средств при проектировании БД затрагивает разные этапы жизненного цикла. Это в определенной мере предопределяет процесс обследования и дает инструмент для отображения его результатов.

Различают прямое проектирование (forward-engineering) - процесс получения структуры базы данных для выбранной целевой СУБД на основе построенной ER- модели, и обратное проектирование (reverse - engineering – реверс - инжиниринг) - когда ER-модель получается на основе существующей базы данных. CASE - средства обычно поддерживают оба эти процесса. Для исследования использовались инструменты AllFusion ERwin Data Modeler 4.1.4.

В данной статье не ставилась цель представить логические и физические модели СУБД. Здесь отмечаются наиболее важные аспекты, которые при моделировании СУБД должны учесть чрезвычайную важность и сложность БД для ситуационного центра в условиях возможного внешнего воздействия на БД при ЧС. Особого внимания, заслуживают вопросы обеспечения высокой надежности и минимума рисков при предупреждении и ликвидации последствий ЧС.

Прежде всего, в ER моделях для **повышения надежности** (в соответствии со стандартом ISO 9126 – одним из важнейших показателей качества БД) следует учесть различные виды сбоев БД. В общем случае для БД известны два вида аппаратных сбоев: «мягкие» сбои, которые приводят к внезапной остановке работы компьютера (например, аварийное выключение питания), и «жесткие» сбои, характеризуемые потерей информации на носителях внешней памяти. Программные сбои — это аварийное завершение работы системы управления базой данных (СУБД) или аварийное завершение пользовательской программы, в результате чего некоторая транзакция остается незавершенной.

Аппаратные сбои фиксируются в каждом конкретном случае в журналах (archive logs) (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. Ведется статистический и системный анализ показателей надежности БД и соответствующих рисков нарушения живучести БД в условиях ЧС.

При этом СУБД должна обладать способностью восстановления последнего согласованного состояния БД после любого аппаратного или программного сбоя.

Самая простая процедура обеспечения надежности восстановления БД — откат транзакции, выполненной пользователем, для чего все записи от одной транзакции связывают обратным списком от конца к началу (аналог Undo).

При «мягком» сбое во внешней памяти основной части БД могут находиться объекты, модифицированные транзакциями, не закончившимися к моменту сбоя, и могут отсутствовать объекты, модифицированные транзакциями, которые к моменту сбоя успешно завершились (по причине использования буферов оперативной памяти, содержимое которых при «мягком» сбое пропадает). В таком случае во внешней памяти журнала должны обязательно находиться записи, относящиеся к операциям модификации обоих видов объектов. Для восстановления БД после жесткого сбоя используют журнал и архивную копию БД.

Физическое разнесение экземпляров БД дает возможность избежать аппаратного сбоя и повышает надежность работы ситуационного центра в целом.

Универсальным принципом обеспечения надёжности является **резервирование**. В общем случае применяются четыре основных вида **резервирования**:

- Аппаратное резервирование, например, дублирование;
- Информационное резервирование, например — методы обнаружения и коррекции ошибок;
- Временное резервирование, например, методы альтернативной логики;
- Программное резервирование, применение независимых функционально равноценных программ.

За основной подход в повышении надежности БД можно использовать резервное копирование, при котором следует размещать на отдельных устройствах резервного копирования. В противном случае при сбое устройства, содержащего БД, резервные копии окажутся недоступными. Кроме того, размещение данных и их резервных копий на отдельных устройствах повышает производительность ввода-вывода, как при записи резервных копий, так и *в процессе производственного использования базы данных* (Рис. 4).

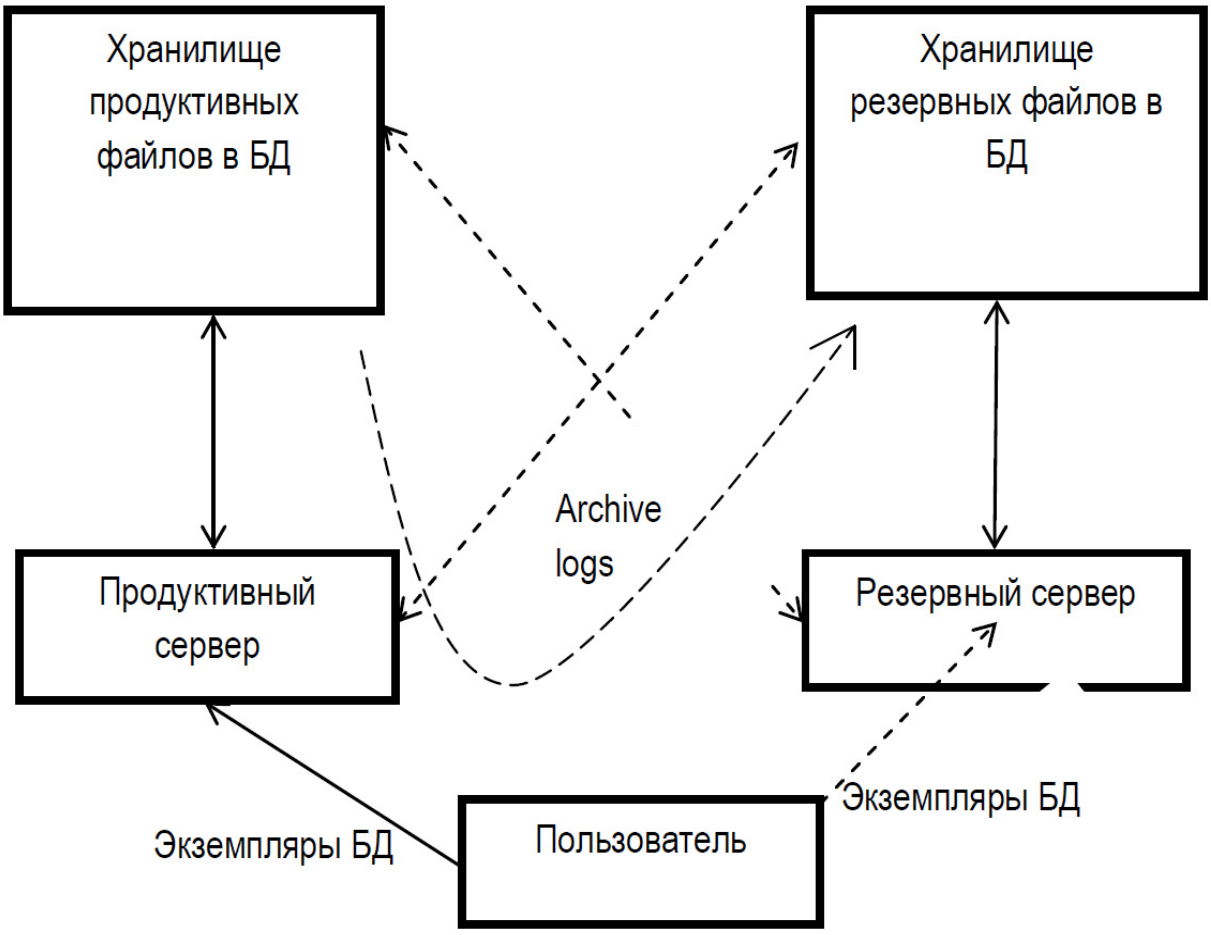

Рисунок 4. Схема резервирования сервера и БД

При логическом и физическом ER моделировании необходимо учесть технические решения по защите данных и сервисов, для ситуационного центра, как правило, важными являются две категории:

- Время восстановления (recovery time objective (RTO)) — допустимое время простоя сервиса в случае сбоя;
- Точка возврата (recovery point objective (RPO)) — допустимый объем возможных потерь данных в случае сбоя.

Эти категории обусловлены задачами ситуационного центра, и от их значений напрямую зависит архитектура предлагаемого решения и, соответственно, его стоимость.

При разработке сервера БД существуют **две проблемы**:

A. Ограничение пропускной способности чтения данных.
B. Ограничение пропускной способности записи данных.

Нет возможности решать две проблемы одновременно. Прежде всего, решается проблема, связанная с чтением данных, когда СУБД не в состоянии обеспечить то количество выборок, которое требуется.

Проблема А – чтения данных решается в такой последовательности:

1. Оптимизируются запросы и конфигурационные параметры (**оптимизация mysql**, **оптимизация postgres**);
2. В целях повышения управляемости, производительности и доступности для больших БД выполняется **секционирование** (англ. *partitioning*) — разделение хранимых объектов БД (таких как таблиц, индексов, материализованных представлений) на отдельные части с раздельными параметрами физического хранения.

Возможные критерии разделения данных, используемые при секционировании:

- по предопределённым диапазонам значений;
- по спискам значений;
- при помощи значений хэш-функций.

Скорее всего, у клиента есть несколько огромных таблиц (обычно всю нагрузку обеспечивают всего несколько таблиц СУБД из всех имеющихся). Причем чтение в большинстве случаев приходится только на самую последнюю их часть (т.е. активно читаются те данные, которые недавно появились). Разработчик распределяет нагрузку на таблицу по ее партициям. Следовательно, выборка типа «***SELECT * FROM articles ORDER BY id DESC LIMIT 10***» будет выполняться только над последней партицией, которая значительно меньше всей таблицы.

3. Следующий пункт разработки возможен **при использовании нескольких серверов**. Это **репликация. Репликация** - это синхронное/асинхронное копирование данных с ведущих серверов на ведомые (или возможно тоже ведущие) сервера. Ведущие сервера называют ***мастерами*** (master), ведомые — ***слейвами*** (slave). Вариантов репликации бывает несколько, но для решения проблемы чтения, используется master-slave репликация. Это решение потребует некоторых изменений в приложении (обычно не больших, хотя не все так просто), т.к. нужно будет читать данные с разных серверов БД, а не одного. Также, необходимо будет учесть репликационный лаг (задержка копирования данных на слейв с мастера — т.е. время, через которое данные полностью скопируются) в работе приложения. Можно выбрать один из двух

вариантов — синхронную или асинхронную репликацию. В случае первой не придется заботиться о лаге, но это отразится на скорости отработки запросов на изменение/вставку данных. Репликация — это наращиваемое решение. Если одного слейва не хватает — ставится второй, третий и т.д. Принцип выбора слейвов на уровне приложения значения не имеет. Главное — это балансирование нагрузки на все слейвы. Естественно существует предельный максимум слейв-серверов (и обычно он связан с тем, что на каком-то этапе уже мастер становится слабым звеном, и начинаются проблемы с записью).

Если были опробованы выше перечисленные способы, то используют **шардинг** — разделение данных на уровне ресурсов. Концепция шардинга заключается в логическом разделении данных по различным ресурсам исходя из требований к нагрузке. Рассмотрим пример. Пусть у нас есть приложение с регистрацией пользователей, которое позволяет писать друг другу личные сообщения. Допустим, оно очень популярно и много людей им пользуются ежедневно. Естественно, что таблица с личными сообщениями будет намного больше всех остальных таблиц в базе (скажем, будет занимать 90% всех ресурсов). Зная это, мы можем подготовить для этой (только одной!) таблицы выделенный высокопроизводительный сервер, а остальные таблицы оставить на других менее производительных. Теперь мы можем идеально подстроить сервер для работы с одной специфической таблицей, постараться уместить ее в память, возможно, дополнительно партиционировать ее и т.д. Такое распределение называется ***вертикальным шардингом***. Если таблица с сообщениями стала настолько большой, что даже выделенный сервер под нее одну уже не спасает. Необходимо делать ***горизонтальный шардинг*** — т.е. разделение одной таблицы по разным ресурсам. На разных серверах будет таблица с одинаковой структурой, но разными данными. Для рассматриваемого примера с сообщениями, мы можем хранить первые 10 миллионов сообщений на одном сервере, вторые 10 — на втором и т.д. необходимо иметь критерий шардинга — какой то параметр, который позволит определять, на каком именно сервере лежат те или иные данные.

Обычно, в качестве параметра шардинга выбирают ID пользователя (user_id) — это позволяет делить данные по серверам равномерно и просто. В предлагаемом примере при получении личных сообщений пользователей алгоритм работы будет такой:

1. Определить, на каком сервере БД лежат сообщения пользователя исходя из user_id;
2. Инициализировать соединение с этим сервером;
3. Выбрать сообщения.

Первый этап – определения конкретного сервера можно решать двумя путями:

1. Хранить в одном месте хеш-таблицу с соответствиями «пользователь=сервер». Тогда, при определении сервера, нужно будет выбрать сервер из этой таблицы. В этом случае узкое место — это большая таблица соответствия, которую ***нужно*** хранить в одном месте. Для таких целей очень хорошо подходят **базы данных «ключ=значение»**;
2. Определять имя сервера с помощью числового (буквенного) преобразования. Например, можно вычислять номер сервера, как остаток от деления на определенное число (количество серверов, между которыми Вы делите таблицу). В этом случае узкое место — это проблема добавления новых серверов. Необходимо делать перераспределение данных между новым количеством серверов.

Для шардинга не существует решения на уровне известных платформ, т.к. это весьма специфическая для отдельно взятого приложения задача. Естественно, делая горизонтальный шардинг необходимо ограничить возможности выборок, которые требуют пересмотра всей таблицы (например, последние посты в блогах людей будет достать невозможно, если таблица постов шардится). Такие задачи придется решать другими подходами. Например, для описанного примера, можно при появлении нового поста, заносить его ID в общий стек, размером в 100 элементом.

Горизонтальный шардинг имеет одно явное преимущество — он бесконечно масштабируем.

Проблему В – **записи данных** следует решать в такой же последовательности, как и проблему чтений. Для начала необходимо рассмотреть возможность партиционирования, далее — репликации (в этом случае мастер-мастер репликация), ну и конечно шардинг.

В большинстве случаев, хватает решений с партиционированием либо репликацией для того, чтобы справиться с требуемыми нагрузками. Современные сервера среднего уровня — это весьма мощные компьютеры, способные выдерживать большие нагрузки при грамотном распределении ресурсов. В уникальных ситуациях Вам придется использовать шардинг. В любом случае, необходимо закладывать возможность гибкого перехода на репликационное решение либо шардинг. Можно применять ***гибридные решения*** и вынести таблицу на отдельный сервер и настроить на нем репликацию.

Необходимо подчеркнуть, что разработка модели управления БД должна соответствовать ИСО/Мэк ТО 10032 –2007.

Оценка рисков и надежности БД

При выполнении этого этапа проектирования для ситуационного центра необходима априорная статистическая информация. Так, например, согласно результатам автоматизированного статистического исследования различного программного обеспечения на предмет ошибок, в исходном коде PostgreSQL было найдено 20 проблемных мест на 775 000 строк исходного кода (в среднем, одна ошибка на 39 000 строк кода). Для сравнения: MySQL — 97 проблем, одна ошибка на 4000 строк кода; FreeBSD (целиком) - 306 проблем, одна ошибка на 4000 строк кода; Linux (только ядро) — 950 проблем, одна ошибка на 10 000 строк кода.

Статистические данные используются для оценки риска при проектировании БД:

$$\Re = \frac{n}{N},$$

где $n-$ количество ошибок с нежелательными последствиями; $N-$максимально возможное их количеству за конкретный период времени.

Эта формула позволяет рассчитать размеры общего и группового риска. При оценке общего риска величина N определяет максимальное количество всех отказов, а при оценке группового риска - максимальное количество отказов в конкретной группе, которая выбрана из общего количества по определенному признаку.

В (Табл. 1) приведены возможные источники увеличения риска и надежности БД при отказах и возможные пути их нейтрализации.

Таблица 1. Возможные источники увеличения риска и надежности БД при отказах и возможные пути их нейтрализации

N п/п	Возможные источники увеличения риска и надежности БД при отказах	Пути нейтрализации отказов
	Ошибочное восстановление БД, для которой выполняется транзакция.	Присвоения каждой базе статуса ошибки для запрашиваемого действия.
	Возникновение временных рисков: потеря данных во время восстановления базы данных (БД может находиться в состоянии транзакций, несмотря на нанесенный ущерб); некоторые таблицы могут быть заполняемы даже во время, когда большая часть информации повреждена.	Ввести экстренный график для снятия TRN LOG резервных копий во время между запросом на восстановление и полным восстановлением.
	Более чем одна ошибка базы данных в один и тот же момент времени.	Необходимо четкое ранжирование ответов на статус базы данных и четкая логика принятия решений о действии по отношению к той или иной БД в зависимости от наиболее верного определенного состояния последней.

Оценивание допустимых значений критериев выбора технического решения

Оценивание качества БД необходимо проводить в соответствии с требованиями стандарта ISO 9126 путем применения шкал количественных и/или балльных оценок свойств или уровней качества БД с несколькими градациями. В (Табл. 2) приведены наиболее важные для решаемой проблемы шкалы оценивания характеристик БД.

Таблица 2. Характеристики качества БД

Характеристики качества	Мера	Шкала
Надежность		
– *Завершенность:*		
• наработка на отказ при отсутствии рестарта.	Часы	10 - 1000
– *Устойчивость:*		
• наработка на отказ при наличии автоматического рестарта;	Часы	10 - 1000
• относительные ресурсы на обеспечение надежности и рестарта.	%	10 - 90
– *Восстанавливаемость:*		
• длительность восстановления.	Минуты	10^{-2} - 10.
– *Доступность-готовность:*		
• относительное время работоспособного функционирования	Вероятность	0,7 - 0,99

Возможный результат проектирования при заданных ограничениях на допустимые значения риска, надежности и показателей качества БД

Полученное техническое решение в результате проведенного исследования представляется системой, которая позволяет содержать 14 резервных копий либо как минимум 2 за последние 2 недели резервных копий типа «FULL». Разработанный алгоритм архивирования копий позволяет добиться результатов с коэффициентом 1:11 от базового размера резервной копии. Центральная БД сообщается с локальными (либо удаленными, но входящими в общую доменную группу) серверами через Push, Pull либо комбинированный метод в зависимости от степени важности БД и возможных временных рамок потерь данных (Рис. 5). Центральная БД служит хранилищем и выполняет управление архивами и резервными копиями БД. Pull метод используется для получения информации от не критических БД. Push метод используется для запросов от критических БД. Комбинированный метод Pull/Push (1 + 1) для критических (сверхкритических) БД. Применяется архивирование Резервных Копий БД.

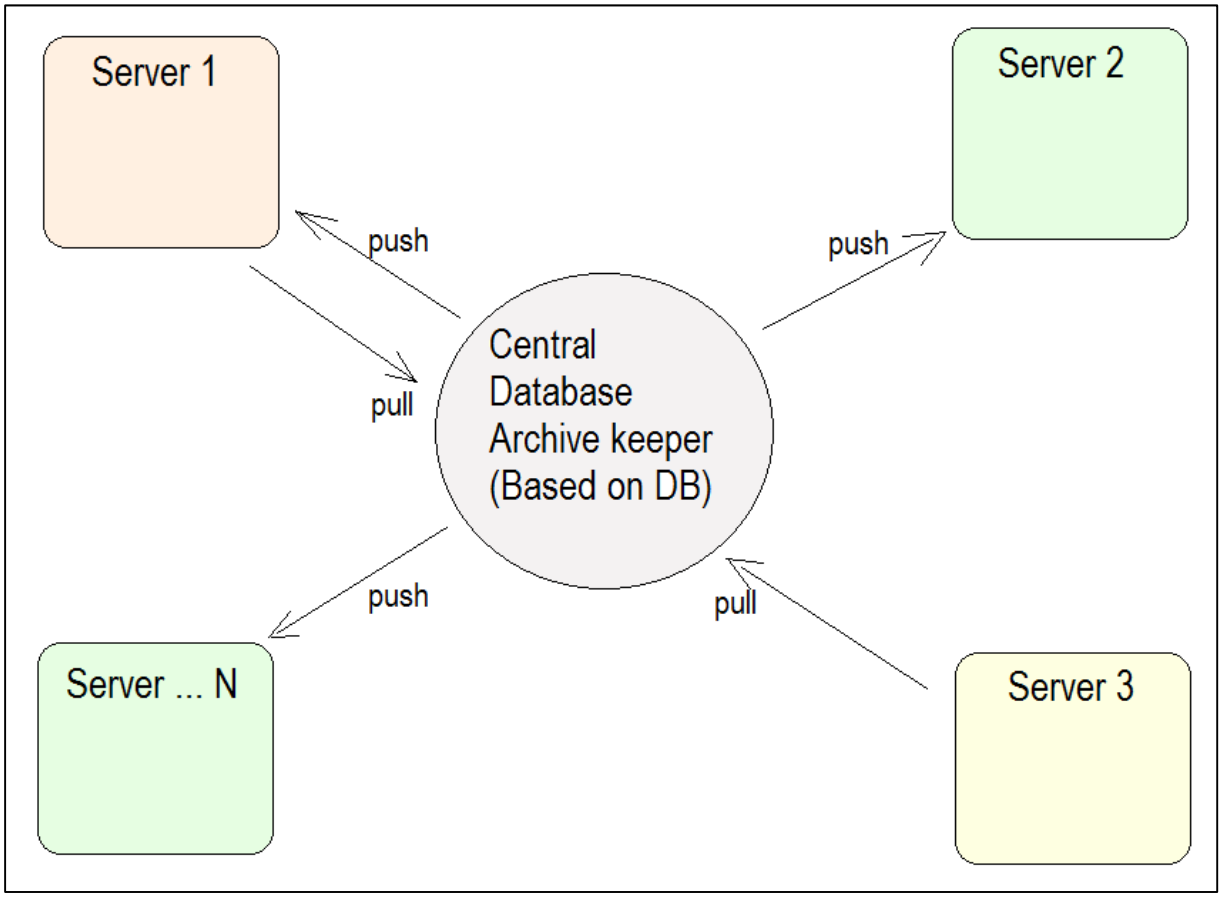

Рисунок 5. Базовые принципы связи основного «хранилища» с серверами

В результате выполнения предлагаемой технологии БД находятся в FULL и DIFFERENTIAL моделях восстановлений. Используется от 1 до N серверов. Для каждой FULL модели восстановления имеется одна последняя архивированная копия на локальном сервере, кроме этого перемещается последняя архивированная копия в центральное хранилище. Максимальное число копий в центральном хранилище

не более 14 (либо за 2 последние недели, в случае если отрезок времени между FULL копиями превышает один день). Обеспечено резервное копирование (Full, Differential, Transaction Log, объекты БД). Используются архивирование копии данных. Применяется трансфер архива (или резервной копии в случае TRN LOG) на центральное хранилище. Выполняется управление архивом резервной копии (помещение в библиотеку, управление мгновенного доступа и подачи на восстановление). В режимах умеренного и полного риска применяется мгновенное восстановление.

Выводы

В статье предложена технологическая последовательность разработки БД для ситуационных центров в условиях возникновения ЧС. Реализация отказоустойчивого решения БД средствами дисковых массивов с применением вышеперечисленных технологий позволяет получить Заказчику необходимую надёжность и достичь поставленной цели.

Обеспечивается быстрое перемещение для SQL БД инстанций на сервере (либо самого сервера, либо группы серверов), быстрое автоматическое восстановление БД на сервере или группе серверов, восстановление данных может быть произведено следующими способами:

- Physical-to-Physical (P2P);
- Physical-to-Virtual (P2V).

Литература

[Kuzemin & Lyashenko, 2007] Alexander Kuzemin, Vyacheslav Lyashenko. Probabilistic and Multivariate Aspects of Construction of the Models and Procedures for Prediction of the Avalanche-Dangerous Situations Initiation. // Proc. of the Second International Conference i.TECH, Sofia, Bulgaria, ITHEA, 2007, pp. 284–289.

[Kuzemin et al, 2007] Alexander Kuzemin, Olesya Dyachenko, Darya Fastova Information Supply of Geo-information Systems for the Forecasting Problem of the Avalanche Danger. // Proc. of the Second International Conference i.TECH, Sofia, Bulgaria, ITHEA, 2007, pp. 289 – 294.

[Kuzomin & Torojev, 2006] Kuzomin A., Torojev A. Mobile means of control and prediction of avalanche climate using information conversion in acoustic. RANGE 291 // IDRC, DAVOS, 2006, Vol. 2, pp. 291 – 294.

[Диго, 2005] Диго С.М. Базы данных: проектирование и использование: Учебник. М.: Финансы и статистика, 2005.

[Куземин & Сорочан, 2004] Куземин А.Я., Сорочан М.В. Понятийное представление ситуации при поиске и классификации проектных решений. //Прикладная радиоэлектроника, Харьков: ХНУРЭ, 2004, Том 3 №3, С. 60 – 67

[Липаев, 2001] Липаев В.В. Выбор и оценивание характеристик качества программных средств. М.: «Синтег», 2001.

Благодарности

Статья публикуется при финансовой поддержке в рамках проекта ITHEA XXI Института информационной теории и приложений FOI ITHEA (*www.ithea.org*) и Ассоциации разработчиков и пользователей интеллектуальных систем ADUIS Украины (*www.aduis.com.ua*).

Информация об авторах

Oleksii Vasylenko *– Aspirant of Kharkiv National University of Radio Electronics; Kharkiv, Ukraine; e-mail: ichbierste@gmail.com; tel.: +380 63 841 66 23*

Major Fields of Scientific Research: General theoretical information research, Knowledge Discovery and Engineering, Business Informatics

Oleksandr Kuzomin *– Prof. Dr., Professor of Informatics Department of Kharkiv National University of Radio Electronics; Kharkiv, Ukraine; e-mail: kuzy@daad-alumni.de; tel.: +38(057)7021515*

Major Fields of Scientific Research: General theoretical information research, Decision Making, Emergency Prevention, Data Mining, Business Informatics

Data Loss Minimization in Situation's Centrums Data Bases

Oleksandr Kuzomin, Oleksii Vasylenko

Abstract: *Problems of reducing risks, increasing the safety and reliability and the use of databases (DB) in emergency situations (ES) are considered in the paper. To solve these problems a technological sequence is proposed. To analyze data, the peculiarity presentation of quantitative and qualitative parameters in emergency situations is calculated. The possibility of logical and functional ER modeling CASE technology with a regard to minimize the risk of life in the emergency area is shown. Technical solution of the problem is given.*

Keywords: *Emergency situations, Database (DB), archive, backup, database management system (DBMS), server.*