УДК 025.4.03;002.53.004.65

МЕТОД РАЗДЕЛЕННЫХ ЗАПРОСОВ ДЛЯ УПРАВЛЕНИЯ УДАЛЕННЫМ ДОСТУПОМ К ДАННЫМ В РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

В.Н. Лебедев, Е.А. Курако, В.Е. Москальков, Д.В. Москальков, В.Л. Орлов

Институт проблем управления им. В.А. Трапезникова РАН, г. Москва

Рассмотрена задача управления защищенным доступом к удаленным базам данных при условии обеспечения гибкости запросной системы, безопасности и гарантированности доставки в современных распределенных информационных системах. Предложен метод ее решения, основанный на разделенных запросах, каждый из которых включает в себя фрагмент управления данными и фрагмент управления доступом. Описан механизм действия метода и приведены языковые конструкции фрагментов. Дано сравнение с другими методами.

ВВЕДЕНИЕ

В распределенных системах обработки, хранения и передачи данных часто возникает необходимость не только обеспечения, но и оперативного управления защищенным доступом (ОУЗД) к удаленным базам данных (БД) из различных географически удаленных точек. Эта проблема особенно актуальна для информационных систем российских и международных правоохранительных органов, вертикально-интегрированных и транснациональных компаний.

Для обеспечения ОУЗД наиболее важно:

- организация гарантированной доставки информации в условиях гетерогенной сетевой среды и некачественных каналов связи;
- обеспечение защиты данных в гетерогенной сетевой среде;
- обеспечение ОУЗД к удаленным БД и гибкого развития запросной системы.

Теоретически предполагается, что удаленный доступ обеспечивается механизмами, предоставляемыми непосредственно системами управления базами данных (СУБД) [1—3]. Однако практически возникают, по крайней мере, три проблемы.

Прежде всего, в СУБД, как правило, не включаются средства гарантированной доставки информации, что вызывает трудности организации доступа при некачественных каналах связи [4]. Далее, средства защиты, обеспечиваемые СУБД, и корпоративные средства защиты сложно интегрируются, как технически, так и в смысле финансовых и временных ресурсов. Обычно это предмет отдельной разработки. И, наконец, для обеспечения гибкого и управляемого удаленного доступа к данным подразумевается возможность отправки с рабочих мест произвольных SQL-запросов, что вступает в противоречие с требованиями безопасности. Введение ограничений доступа к данным средствами СУБД для различных рабочих мест, с одной стороны, снижает уровень гибкости, с другой — не решает полностью проблемы защиты, так как семантический контроль при этом практически отсутствует.

Проблема организации гарантированной доставки решается обычно путем разделения задачи обработки на клиентскую и серверную части (метод разделения задач). При этом организация доступа по сети может быть реализована различными способами, что обеспечивает надежность транспортировки. Уровень обеспечения безопасности также может быть повышен, но при такой организации снижается гибкость доступа, так как введение новых запросов требует перепрограммирования клиентских программ.

Если же на клиентском уровне подключить механизм сценариев, которые будут интерпретироваться программами сервера приложений, то проблема гибкости решается за счет снижения степени безопасности по причинам, рассмотренным выше.

В данной статье предлагается новый, апробированный на практике метод решения перечисленных задач — *метод разделенных запросов*.

1. МЕТОД РАЗДЕЛЕННЫХ ЗАПРОСОВ

Общая схема удаленного доступа к данным с помощью метода разделенных запросов представлена на рис. 1.

Подразумевается конструирование запросов в форме сценариев, что обеспечивает добавление в систему новых возможностей в процессе эксплуатации [5].

Однако, как уже отмечалось, использование сценариев снижает уровень безопасности программного комплекса ввиду невозможности точного прогнозирования содержания сценариев, формируемых на рабочих местах.

Для решения этой проблемы все сценарии (запросы) разделяются на фрагмент управления данными и фрагмент управления доступом.

Фрагмент управления данными представляет собой форму ввода параметров запроса, которая размещается на удаленном рабочем месте. Форма ввода имеет формат HTML, что позволяет создавать ее либо на основе ранее подготовленных HTML-шаблонов, либо с помощью HTML-редактора.

Принципиальное отличие от известных методов состоит в том, что данная форма не поступает по сети от web-сервера, а размещается на рабочем месте в локальной файловой системе, где и активизируется. Программное обеспечение, обслуживающее фрагмент управления данными, выбирает введенные оператором параметры и создает блок данных запроса в формате XML [6], который передается в серверный центр. Таким образом, на рабочем месте формируются только параметры (данные) запроса. Более того, с определенного рабочего места могут приниматься и обрабатываться в серверном центре только определенные наборы данных.

Фрагмент управления доступом представляет собой скрипт (интерпретируемую часть сценария), который размещается на сервере приложений. Программное обеспечение обслуживания фрагмента управления доступом обеспечивает прием блоков данных, определяет полномочия пользователя, приславшего тот или иной блок, и в случае положительного решения — формирует SQL-запросы к БД на основе скрипта и блока данных запроса. Результат выполнения запроса — ответ — пересылается на удаленное рабочее место, регистрируется в локальной базе данных и отображается на мониторе.

Модификация сценариев одно из преимуществ метода разделенных запросов заключается в простоте модификации запросной системы, например, при изменении структур данных или требований к запрашиваемой информации. Необходимо лишь обновить оба фрагмента сценария или создать новый сценарий. Фрагмент управления доступом размещается на сервере приложений, а фрагмент управления данными рассылается на удаленные рабочие места, которые должны его использовать. Процесс настройки схематично показан на рис. 2. Отметим, что на рабочем месте новая форма ввода автоматически включается в список поддерживаемых форм.

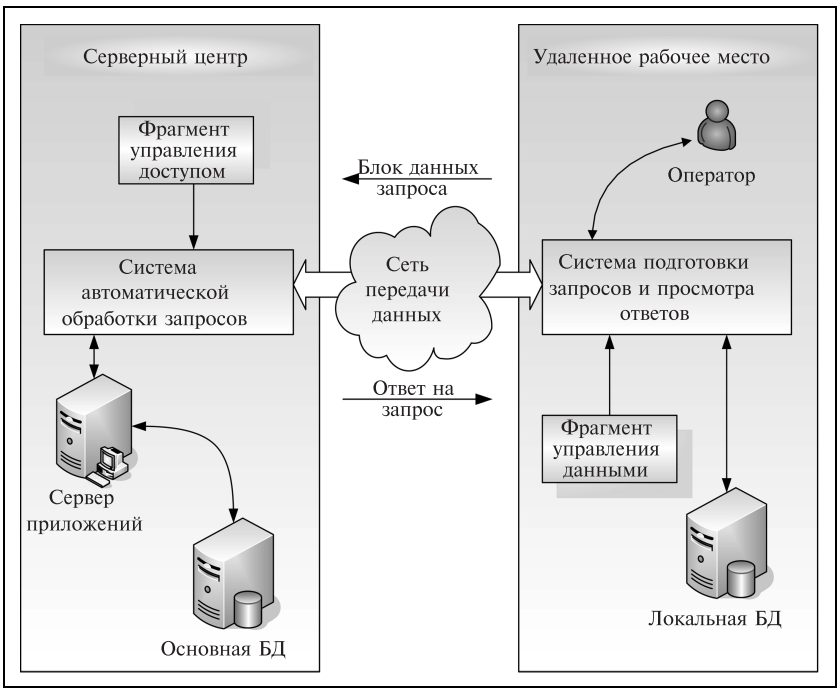

Рис. 1. Схема удаленного доступа к БД с помощью метода разделенных запросов

2. ЯЗЫКОВЫЕ КОНСТРУКЦИИ МЕТОДА РАЗДЕЛЕННЫХ ЗАПРОСОВ

2.1. Фрагмент управления данными

Фрагмент управления данными или форма ввода представляет собой файл в HTML-формате. Каждая форма ввода должна содержать, по крайней мере, один тег FORM, который включает в себя элементы ввода данных (теги INPUT,

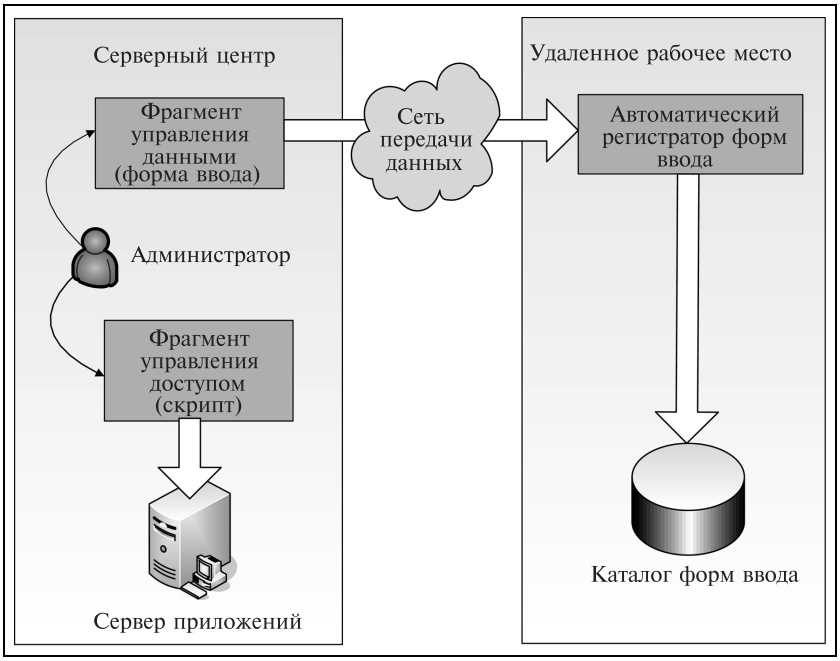

Рис. 2. Добавление новых типов запросов

SELECT и т. д.). Допускается использование в форме ввода различных стилей, скриптов и т. п., например, для контроля вводимых данных.

Для перехода от простой формы ввода к фрагменту управления данными необходимо включить в элемент FORM управляющий список элементов INPUT с атрибутом *hidden*. Этот список обеспечивает полноту и корректность формирования *блока данных запроса*, который создается после ввода данных в форму и предназначается для обработки в серверном центре.

Каждый элемент управляющего списка включает в себя имя элемента (атрибут *name*) и его значение (атрибут *value*). Перечень допустимых имен и описание их значений приведены в табл. 1.

В качестве примера приведем управляющий список вида:

```
<INPUT type=hidden name="H_OBJECT"
   value="DocNum,IssuingCountry,DocType">
<INPUT type=hidden name="H_DB"
   value="BASE1">
<INPUT type=hidden name="H_FORMAT"
   value="FDC">
<INPUT type=hidden name="H_OPTION"
   value="OFFLINE">
```

Для контроля корректности ввода данных при заполнении формы используется атрибут *id* каждого элемента ввода данных INPUT типа text. По существу значение *id* представляет собой строку, которая включает в себя перечень интерпретируемых команд контроля ввода данных. Описание каждой команды начинается с символа «$», за которым следует идентификатор команды. За идентификатором в круглых скобках следует список параметров. Фактическим разделителем элементов списка является знак «$». В частном случае список может отсутствовать. Перечень команд:

- $Obl — присутствие этой команды указывает на то, что данное поле ввода обязательно для заполнения;
- $Len(N) — команда контроля максимальной длины вводимых данных;
- $Min(N) — команда контроля минимальной длины вводимых данных;
- $Rus — присутствие этой команды указывает на то, что допускается возможность ввода только русских символов;
- $Lat — при включении этой команды разрешен ввод только латинских символов;
- $Dt([format1]<formatTo>,...,[formatN]<formatTo>) — определение поля как поля ввода даты и указание формата (форматов) даты, например, команда $Dt([dd.mm.yyyy]<yyyy/mm/dd>,[yyyy]) разрешает ввод даты в двух форматах: dd.mm.yyyy и yyyy (день, месяц, год или просто год).

Пример оформления элемента INPUT:

```
<INPUT type=text name="BirthDate"
id="$Dt([dd.mm.yyyy]<yyyy/mm/dd>,[yyyy])"
size=10>
```

Таблица 1

Допустимые имена и их значения в управляющем списке

Имя	Значение
H_OBJECT	Перечень названий полей ввода, значения которых должны отсылаться в серверный центр (разделитель здесь и в дальнейшем — запятая)
H_DB	Перечень идентификаторов баз данных, для которых предназначен запрос
H_FORMAT	Уникальный идентификатор типа запроса, соответствующий идентификатору фрагмента управления доступом, размещенного в серверном центре. Необходим для связывания фрагментов запроса
H_NAME	Текстовое описание данного типа запроса (не обязательно)
H_OPTION	Перечень дополнительных параметров запроса, например, здесь может быть явно указан возможный способ доставки ONLINE и (или) OFFLINE

Блок данных запроса представляет собой файл, содержащий текст следующего вида в формате XML:

```
<?xml version="1.0" encoding="WINDOWS-1251" ?>
<MSG>
   <TO TName="..." />
   <FROM FName="..." />
   <DOCUMENT DocNum="..." DocDate="...">
      <BODY ExecutorCode="..." ExecutorName="..." />
      <REQUEST Type="..." DB_ID="..." Name="...">
         <PARAM Name="..." ForeName="..."/>
         <PARAM Name ="..." ForeName ="..."/>
      </REQUEST>
   </DOCUMENT>
</MSG>
```

Блок данных включает в себя следующие теги:

- MSG — основной тег, определяющий блок данных как сообщение;
- TO — содержит имя серверного центра (атрибут TName);
- FROM — содержит имя удаленного рабочего места (атрибут FName);
- DOCUMENT — содержит служебную информацию для регистрации запроса в БД на серверном центре (атрибут DocNum — регистрационный номер исходящего документа, DocDate — дата и время отправки в формате дд.мм.гггг чч:сс); эта информация формируется динамически во время отправки запроса с удаленного рабочего места;
- BODY — содержит информацию об операторе, который непосредственно отправляет данный запрос (атрибут ExecutorCode — код оператора (необязательный параметр), ExecutorName — имя оператора);
- REQUEST — содержит информацию о запросе (его тип — Type, название — Name и идентификатор БД — DB_ID). Эти данные берутся из форм ввода (фрагментов управления данными).

Тег REQUEST может содержать внутри себя произвольное число тегов PARAM, каждый из которых имеет набор атрибутов, которые должны быть использованы при поиске информации в БД. Наименования атрибутов должны соответствовать наименованиям полей ввода, используемым во фрагменте управления данными, а значения — значениям этих полей, которые ввел оператор при заполнении формы ввода. Эти же имена полей используются во фрагменте управления доступом, обеспечивая связь этого фрагмента с фрагментом управления данными.

В теге REQUEST допускается использование списков. Для этого в состав блока данных включается несколько тегов PARAM. В этом случае поиск в БД должен вестись по всему перечню наборов данных, и в ответе должен быть представлен суммарный результат.

2.2. Фрагмент управления доступом

Синтаксис фрагмента управления доступом базируется на синтаксисе SQL-запросов, но требует включения в текст следующих расширений.

- Каждое поле из перечня, следующего за оператором SELECT, должно иметь подполе комментария. Комментарий должен представлять собой описание данного поля. Если для отображения значения данного поля требуется подключение справочника, то после описания в квадратных скобках указывается наименование данного справочника.
- Если в структуры оператора WHERE требуется поместить какое-либо значение, то вместо этого значения указывается соответствующее имя атрибута тега PARAM, присутствующего во фрагменте управления данными. Имя должно быть заключено в квадратные скобки.
- В случае использования справочников требуется непосредственно за модифицированным SQL-запросом разместить оператор REFERENCE, за которым должен следовать перечень строк. Каждая строка должна состоять из наименования справочника, указанного в поле комментария (см. выше) и следующего за ним SQL-запроса, который обеспечит выборку описания поля по значению его кода.

Таким образом, в качестве значений полей, по которым ведется поиск, задаются не статические величины, а переменные, значения которых подставляются из блока данных, т. е. формируется параметрический SQL-запрос. Пример запроса поиска данных по условной базе данных фирм:

```
SELECT
   OBJECT_NUM,     /* Идентификатор */
   FIRM_TYPE,      /* Тип Фирмы [Типы фирм]*/
   NAME_RUS,       /* Название */
   REG_ADDR,       /* Адрес регистрации */
   FIRM_INN,       /* ИНН */
   FIRM_OPEN_DATE, /* Дата Регистрации */
FROM
   OBJECT_REG R,
   FIRM F,
WHERE
   F.ID_FIRM=R.OBJECT_NUM AND
   R.OBJECT_ID = 'FRM' AND
   F.NAME_RUS like '[FirmName]' AND
   F.FIRM_TYPE='[FirmType]'
REFERENCE
Типы фирм SELECT FIRM_TYPE,FIRMT_NAME
FROM FIRM_TYPE_VOC
```

2.3. Ответы на запросы

Ответы на запросы, которые формируются в серверном центре как файлы и отсылаются на удаленное рабочее место, также имеют XML-формат.

Синтаксическая конструкция ответа имеет следующий вид:

```
<?xml version="1.0" encoding="WINDOWS-1251" ?>
<ANSWER>
<INFO DB_ID="..." Req="..."/>
<COMMENT>Найдено объектов: 1</COMMENT>
<TABLE>
<FORMAT>
   <COL Name="поле1" Value="FIELD1"/>
   ...
   <COL Name="полеN" Value="FIELDN"/>
</FORMAT>
<STR FIELD1="значение1-1" ... FIELDN="
значение1-N" />
...
<STR FIELD1="значениеM-1" ... FIELDN="
значениеM-N" />
</TABLE>
</ANSWER>
```

Используемые теги:

- ANSWER — основной тег;
- INFO — тег информации о запросе, на который сформирован ответ, имеет атрибуты:
  - DB_ID — идентификатор БД, из которой проводилась выборка информации;
  - Req — идентификатор запроса, соответствующий атрибуту DocNum тега DOCUMENT в блоке данных запроса;
- COMMENT — тег комментариев, предназначен для вывода дополнительной информации оператору на удаленном рабочем месте;
- TABLE — тег, включающий в себя данные ответа, которые описываются тегами FORMAT и STR;
- FORMAT — содержит описание столбцов таблицы, причем каждый столбец описывается тегом COL, который определяет столбец таблицы и содержит следующие атрибуты:
  - Name — наименование столбца (будет отображаться в заголовке таблицы);
  - Value — наименование соответствующего атрибута тега STR;
- STR — имеет набор атрибутов, имена которых являются идентификаторами полей ответа, а значения — данными ответа.

3. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МЕТОДА

Метод разделенных запросов программно реализован и применен в территориально распределенной автоматизированной информационной системе, функционирующей в гетерогенной сетевой среде и основанной на трехзвенной архитектуре (клиент — сервер приложений — сервер БД). В ходе тестирования и последующей эксплуатации системы были получены результаты, подтверждающие гибкость метода в сочетании с обеспечением гарантированной доставки данных и требуемым уровнем информационной безопасности. Результаты экспертного сравнения предложенного метода с известными методами удаленного доступа приведены в табл. 2. Каждый из методов оценивался на основании экспертных данных по трем показателям качества решения перечисленных во Введении задач (гибкость, защищенность, гарантированность) и трехбалльной шкале (0 — показатель не обеспечивается, 1 — обеспечивается, 2 — расширенные возможности). В табл. 2 даны средние арифметические значения полученных оценок. Видно, что метод разделенных запросов по этим показателям предпочтителен.

Таблица 2

Сравнение методов удаленного доступа

Метод доступа	Гибкость системы	Гарантированность доставки	Защищенность системы
Метод разделенных запросов	1,86	1,57	1,86
Доступ к данным средствами СУБД	1,43	1	1
Метод разделения задач	0,29	1,57	1,86

ЗАКЛЮЧЕНИЕ

Предложенный метод может быть применен во всех случаях, когда требуется обеспечить управляемый защищенный доступ к удаленным БД из территориально распределенных рабочих мест как в однородной, так и в гетерогенной сетевой среде. Он обеспечивает надежную транспортировку данных, требуемый уровень информационной безопасности и гибкость при модернизации и развитии системы. Метод разделенных запросов программно реализован и внедрен в системе удаленного доступа к БД Национального центрального бюро Интерпола при МВД России и его региональных филиалов.

ЛИТЕРАТУРА

1. *Арсеньев Б.П., Яковлев С.А.* Интеграция распределенных баз данных. — М.: Лань, — 2001.
2. *Дейт К.Дж.* Введение в системы баз данных. — М.: Вильямс, 1999.
3. *Базы* данных. Интеллектуальная обработка информации / В.В. Корнеев, А.Ф. Гареев, А.Ф., С.В. Васютин, В.В. Райх. — М.: Нолидж, — 2000.
4. *Асратян Р.Э., Орлов В.Л., Шинкарюк А.Г.* Единый связной интерфейс. Тр. ПУ РАН. — 2000. — Т. IX.
5. *Москальков Д.В.* Организация защищенного доступа к удаленным базам данных. Тр. XIII междунар. конф. «Проблемы управления безопасностью сложных систем», Москва, декабрь 2005 г., 261—263 стр., М.:РГГУ, 2005.
6. *XML.* Новые перспективы WWW / Ф. Бумфрей, О. Диренцо, Й. Дакетт и др. — М.: ДМК, 2000.

☎ *(495) 334-92-81, e-mail: lebvini@pui.ru*

Статья представлена к публикации членом редколлегии В.Л. Эпштейном. ❑