О ДОСТУПЕ К РЕЛЯЦИОННЫМ СУБД ПО ПРОТОКОЛУ Z39.50

Жижимов О.Л. (zhizhim@uiggm.nsc.ru), *Скибин С.В.* (skibin@uiggm.nsc.ru)
ОИГГМ СО РАН, Новосибирск,
проспект ак. Коптюга, 3

Аннотация

В докладе разбирается вариант построения системы “сервер Z39.50 + сервер реляционной СУБД”: специфика обработки запросов, технология приведения реляционных данных к форматам Z39.50. Рассматриваются сложности, возникающие при отображении реляционной модели данных на иерархическую модель данных Z39.50. Анализируются различные варианты представления структурированной информации в Z39.50: GRS-1, SQL-RS, XML. Доклад основывается на опыте авторов, полученном при разработке сервера Z39.50 ZooPARK. Прототип системы доступен по адресу http://z3950.uiggm.nsc.ru/zgwk/kadrs.htm.

Введение

Существует множество проектов, которые тем или иным способом включают в себя работу с базами данных. В настоящем докладе будут затронуты только открытые информационные системы. Очень часто каждая такая система работает со своими уникальными базами данных, что подразумевает использование различных СУБД и различных схем данных. При этом остается актуальным вопрос о взаимодействии различных открытых информационных систем, т.е. о стандартизации сетевой работы с базами данных [3]. Но отказываться от существующей и хорошо работающей системы или вести дублирование баз данных является нецелесообразным и мало приемлемым решением. Поэтому появились различные технологии создания распределенных информационных систем с использованием существующих решений.

Среди множества технологий обеспечения унифицированного сетевого доступа к базам данных на сегодняшний день наиболее проработана технология, основанная на протоколе Z39.50 (ISO23950) [1]. Использование этого протокола позволяет строить распределенную информационную систему, не заботясь о программно-аппаратной платформе и деталях архитектуры каждой конкретной ее подсистемы и используемых в ней СУБД [2].

Для информационных систем, основанных на реляционных СУБД, которые широко используются в мире для организации хранилищ данных и систем оперативной обработки транзакций, существуют альтернативные технологии построения распределенных систем, но эти технологии не удовлетворяют в полной мере современным требованиям [3]. Необходимый для подобного рода технологий уровень описания сетевого взаимодействия и процедур доступа к базам данных наиболее тщательно проработан в Z39.50. Примером системы на основе Z39.50 может служить сервер ZooPARK и программа-клиент, установленная на WWW-шлюзе http://z3950.uiggm.nsc.ru/zgwk/kadrs.htm.

В докладе разбирается вариант построения системы “сервер Z39.50 + сервер реляционной СУБД”: специфика обработки запросов, технология приведения реляционных данных к форматам Z39.50. Анализируются различные варианты представления структурированной информации в Z39.50: GRS-1 [1], SQL-RS[4], XML. Доклад основывается на опыте авторов, полученном при разработке сервера Z39.50 ZooPARK.

Особенности Z39.50 и реляционных СУБД

Рассмотрим кратко основные характеристики Z39.50.

Z39.50 [1] – это стандарт, разработанный ANSI и принятый ISO в 1998 г. (ISO-23950). Стандарт описывает процедуры сетевого доступа к базам данных. Вся идеология Z39.50 построена на абстрагировании от реализации конкретной системы. При этом каждая “физическая” база данных должна быть отображена на абстрактную модель Z39.50, элементы которой описываются в терминах уникальных идентификаторов (OID – идентификатор объекта). Например, существуют такие классы OID:

- APDU (определение структуры сетевых пакетов данных);
- attributeSet (наборы поисковых атрибутов);
- recordSyntax (форматы представления возвращаемых данных);
- schema (схемы данных);
- другие классы OID.

За рамками протокола остается вопрос о способах хранения данных и вопрос о вариантах реализации конкретных систем. Таким образом, для хранения информации могут использоваться различные СУБД. Исторически сложилось так, что основной сферой применения Z39.50 являются библиотечные системы, в которых для хранения информации часто применяются базы данных ISIS, не являющиеся реляционными. Лишь в феврале 2000 г. было принято дополнение к стандарту, которое отдельно описывает особенности использования реляционных СУБД в системах на базе Z39.50.

Реляционные СУБД лидируют по массовости своего распространения в мире, предоставляют эффективные методы хранения данных, а также отличается высокой эффективностью обработки запросов (Oracle, Sybase, Microsoft SQL Server, Informix) [5]. Причем на рынке существуют как высококлассные коммерческие разработки, перечисленные выше, так и свободно распространяемые системы приемлемого качества (MySQL, Postgres). Но каждая СУБД использует на сетевом уровне свои фирменные протоколы общения между клиентами и сервером, что подразумевает наличие специально разработанных для каждой конкретной СУБД программ-клиентов.

Для построения распределенных информационных систем также важен единый стандарт схем данных, который поддерживается протоколом Z39.50. Таким образом, авторам кажется целесообразным описать особенности решений, сочетающих в себе использование Z39.50 и реляционных СУБД. Данное описание особенно актуально в свете современных тенденций к стандартизации в области открытых информационных систем, роста популярности Z39.50 и сравнительной новизны дополнений к протоколу Z39.50, помогающих в работе с реляционными СУБД.

Пример распределенной информационной системы

В информационной системе, показанной на рисунке, сервер Z39.50 ZooPARK является промежуточным слоем между клиентами Z39.50 и серверами баз данных.

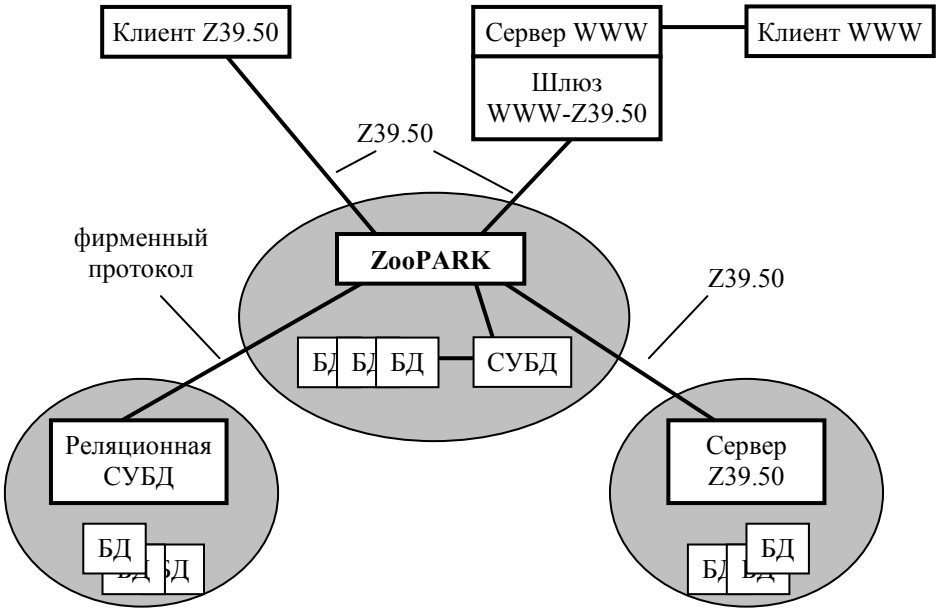


Рис. 1. Схема распределенной информационной системы на основе ZooPARK

Рассмотрим более подробно связку клиент Z39.50 – ZooPARK – реляционная СУБД. Как было отмечено выше, Z39.50 оперирует абстрактными идентификаторами объектов (OID). Любая база данных должна быть отображена на эту модель (в более общем случае несколько физических БД могут быть объединены в одну БД, к которой и обращаются клиенты Z39.50 при взаимодействии с сервером.). Такое отображение подразумевает следующие этапы:

- преобразование запросов из форматов Z39.50 к запросам SQL;
- преобразование полученной информации к схемам данных Z39.50;
- форматирование информации к требуемому клиентом формату данных;
- преобразование русскоязычных кодировок;
- др. операции, такие как объединение информации от нескольких БД, последующая сортировка и т.д.

Каждый из этих этапов имеет свои сложности, свои узкие места. Затронем более подробно первые три ключевых момента именно в связи с использованием в качестве хранилищ информации реляционных СУБД.

Преобразование запросов

Каждая система на основе протокола Z39.50 должна поддерживать как минимум обработку запросов в формате RPN (обратная польская нотация), который выражается при помощи набора абстрактных идентификаторов (цифровых индексов OID) [1]. Сервер Z39.50, получив такой запрос, должен преобразовать его к форме, приемлемой для конкретной базы данных. Сервер Z39.50 может обрабатывать запросы в форме ZSQL, которые также могут опираться на абстрактные структуры данных (например, вместо названия поля таблицы следует использовать его цифровое значение из соответствующего набора атрибутов, а в секции FROM указывать идентификатор схемы данных и т.д.) [4]. Запросы ZSQL могут выражаться и в форме “привычного” SQL, но это противоречит идеологии Z39.50, хотя и может использоваться в узком кругу как альтернатива ODBC, JDBC и т.п. технологиям, предоставляя единый сетевой интерфейс для обмена данными. При использовании реляционных СУБД необходимо определять, какие таблицы будут участвовать в запросе, каков будет тип связей между ними, учесть возможность произвольной вложенности запросов. На рис.2

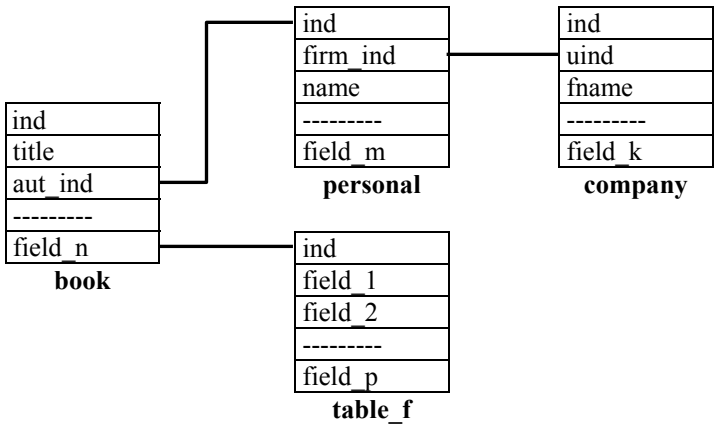
а)

@and @attr 1=4 **@attr** 5=3 {система} **@attr** 1=1005 {ОИГГМ} **@attr** 1=1003 {Жижимов}

б)

select [(2,1)]
from [1.2.840.10003.13.2]
where [(2,4)] = (select min([(2,4)])) from [1.2.840.10003.13.2])

в)



показан пример запроса RPN и ZSQL с использованием OID поисковых атрибутов Bib-1 и меток (tagset-G). Более подробную информацию о запросах можно получить в описании спецификаций протокола [1,4].

Для установления соответствий между реальными таблицами и поисковыми атрибутами в ZooPARK используются статические конфигурационные файлы. В этих файлах, которые могут быть уникальными для каждой базы данных, должна присутствовать информация о соответствии элементов OID элементам SQL-запросов. Если ограничиться только поисковыми запросами (select), то конфигурационные файлы должны содержать информацию о соответствии полей таблиц поисковым атрибутам и элементам схем данных, о типах полей, о характере связей между таблицами. В общем случае если создаются унифицированные преобразователи запросов, то неизбежно возникает потребность в информации о диалектах SQL той или иной СУБД, т.к., несмотря на существование стандартов SQL, в языках запросов имеются некоторые различия [5].

В общем случае структура конфигурационного файла довольно сложна. Но для ряда приложений подобные файлы можно существенно упростить. Ведь очень часто объем и характер информации, предоставляемой открытыми информационными системами, не подразумевает использование сложных запросов ко многим связанным таблицам. Часто бывает разумнее воспользоваться хранимыми процедурами и псевдотаблицами (view) вместо нагрузки на сервер Z39.50.

Рис. 2. Преобразование запросов: а) запрос в форме RPN; б) запрос в форме ZSQL-abstract; в) пример таблиц и связей между ними в реляционной БД, которая содержит искомую информацию

Существует еще одна важная особенность систем “Z39.50 + реляционные СУБД”. Это различная логика обработки запросов. Select-запросу SQL соответствует две фазы запросов Z39.50: search – запрос на количество записей, удовлетворяющих поисковым требованиям, и present – возвращение записей клиенту порциями. Таким образом, необходимо выполнять два вида запросов: первый – чтобы получить информацию о количестве записей, и второй – для извлечения записей. Причем последний запрос может исполняться много раз подряд и здесь важно найти оптимальный способ общения с СУБД. SQL-запросы, соответствующие search и present, могут сильно отличаться как по своей форме, так и по характеру возвращаемой от СУБД информации. Так, в первом случае возможным результатом выполнения запроса будет число записей. Во втором же случае серверу необходимо обработать схему данных, в соответствии с которой информация будет возвращена клиенту Z39.50.

Это может потребовать выполнения целого ряда взаимосвязанных запросов, но об этом более подробно будет сказано далее. Пока же важно то, что при разработке решений на основе Z39.50 необходимо учитывать тонкости, возникающие на этапе преобразования запросов.

Обработка схем данных

Z39.50 оперирует иерархическими схемами данных, а не реляционными. Таким образом, для заполнения требуемой схемы данных может потребоваться произвести несколько взаимосвязанных запросов к базе данных. Но поскольку схемы данных, в отличие от поисковых требований программ-клиентов, заранее известны, существует возможность жестко описать последовательность действий при взаимодействии с базами данных. В сервере ZooPARK для этого применяются специальные конфигурационные файлы в форме, подобной XML, для удобства работы администраторов сервера. Ниже приведен пример такого файла.

```
<persons>
        <Схема> PERSONS-schema </Схема>
        <Номер_локальный> pers_uiggm:$r.ind$ </Номер_локальный>
        <Фамилия> $r.fam$ </Фамилия>
        <Имя> $r.nam$ </Имя>
        <Отчество> $r.otc$ </Отчество>
…
        <$ sql edu select a.*, b.edu_s,doc_name='Диплом'
                        from html.OK_EDU a, html.SPR_EDU b
                        where a.ind=$r.ind$  and a.edu_cod>0 and a.edu_cod=b.edu_cod $>
        <Образование>
                <$ for edu $>
                <Учебное_заведение>
                        <Название> $edu.nam_vuz$ </Название>
                        <Дата_окончания> $edu.g_fin_v$ </Дата_окончания>
                        <Специальность> $edu.spc_edu$ </Специальность>
                        <Факультет> $edu.facultet$ </Факультет>
                        <Тип> $edu.edu_s$ </Тип>
                        <Документ>
                                <Название> $edu.doc_name$ </Название>
                                <Серия> $edu.ser_diplm$ </Серия>
                                <Номер> $edu.num_diplm$ </Номер>
                                <Дата_выдачи> $edu.dat_f_dipl$ </Дата_выдачи>
                        </Документ>
                </Учебное_заведение>
        </Образование>
        <$ close edu $>
</persons>
```

На этом примере показано, как можно заполнять схемы данных. Жирным шрифтом выделены команды выполнения и обработки вложенных запросов. Сервер ZooPARK использует эти команды при обработке соответствующих схем данных. Информация после выполнения вложенного запроса подставляется в

конкретные позиции в соответствии с шаблоном. В результате получается структура, соответствующая конфигурационному файлу-шаблону, но содержащая реальную информацию. На следующем этапе информация преобразуется к требуемому формату представления данных. Понятно, что не всякий формат может быть использован в общем случае. Например, нельзя передать посредством текстового формата графическую информацию, а при помощи формата jpeg – текст. Внутренний формат должен быть достаточно универсальным и вместе с тем нести в себе минимум избыточной информации.

Форматы представления данных

В Z39.50 определен широкий спектр допустимых форматов, в которых данные от сервера передаются клиенту. Форматы условно можно разделить на простые, или визуальные (текст, HTML, BMP и т.п.), и сложные, передающие структуру данных (структурированные). Именно последняя группа заслуживает пристального внимания при построении открытых информационных систем. В этой группе форматов рассмотрим следующие: GRS-1, SQL-RS, XML, каждый из которых имеет свои плюсы и минусы.

GRS-1 широко распространен в среде Z39.50, являясь стандартом де-факто для передачи структурированных данных. Его поддерживает большинство существующих приложений Z39.50. Этот формат определяет иерархическую структуру записи. Отображение в нем данных реляционных таблиц и, тем более, передача дополнительной метаинформации о таблицах связана с некоторыми трудностями. SQL-RS, напротив, был разработан для связки “Z39.50 + реляционная СУБД” и предоставляет максимум возможностей при доступе к реляционным данным через Z39.50. Основной недостаток этого решения – слабая поддержка со стороны разработчиков приложений Z39.50, особенно клиентских программ. Такая ситуация в первую очередь связана со сравнительной молодостью SQL-RS, окончательные спецификации которого появились лишь в начале 2000 г. Другой причиной является некоторое его противоречие одному из принципов Z39.50 – “использование стандартных схем данных, иерархических структур записей и форматов, отображающих эти схемы”.

Еще один формат, заслуживающий внимания, – XML – в отличие от GRS-1 и SQL-RS не является “родным” для Z39.50, но в приложениях WEB получил большое признание в мировом сообществе. Основные достоинства XML – простота и читабельность без предварительной обработки. Эти плюсы неизбежно ведут за собой и минусы: сравнительно большой объем накладных расходов сформированных тегов, передача только текстовой информации. К минусам стоит отнести и относительное неудобство при дальнейшей обработке информации: текстовый поиск тегов, проблемы с типами данных, связями и т.д., что требует дополнительной стандартизации.

Форматирование записей

Формат внешнего представления записей определяет программа-клиент Z39.50. Поэтому в сервере ZooPARK на последней стадии происходит преобразование к требуемому клиентом формату. А все промежуточные преобразования, такие как обработка схем данных, смена кириллических кодировок для текстовой информации, производятся на удобном внутреннем формате. Сервер ZooPARK поддерживает все приведенные выше структурированные форматы, а также простые, такие как HTML, SUTRS и др. Безусловно, выбор того или иного набора форматов в серверах и клиентах определяется, в конечном счете, его популярностью как среди разработчиков, так и среди пользователей. Процесс внесения нового формата заключается просто в настройке схемы преобразования из структурированных форматов в новый. (Лишь в критичных приложениях, таких как аудио-видео, будет требоваться особая проработка.) Поэтому мы не будем

подробно останавливаться на преобразовании форматов в ZooPARK, но перечислим наши критерии выбора тех или иных форматов.

Во-первых, для большинства приложений вполне достаточно структурированной текстовой информации, такой как HTML, XML. Более того, при использовании, например, HTML появляется возможность отображения графики по ссылкам. Внутренний формат сервера при этом должен содержать минимум лишней информации, такой как типы полей. Для более серьезных систем поддерживаются GRS-1 и SQL-RS. Причем GRS-1, по сути, является единственным универсальным форматом, пригодным для построения распределенных информационных систем на основе Z39.50. Использование SQL-RS было бы уместно при разработке протоколов общения с реляционными СУБД напрямую. Если нет необходимости знать структуру таблиц и характер связей между ними, формат GRS-1 является предпочтительней. Так, пока только единичные разработки серверов Z39.50 в какой-то мере обрабатывают ZSQL и SQL-RS. Клиентские разработки в данном направлении пока практически отсутствуют.

XML удобен в тех случаях, когда передаются небольшие блоки структурированной текстовой информации, т.е. для создания интернет-приложений, направленных на общение с человеком, что соответствует приоритетному направлению развития XML.

Интересным представляется тот факт, что преобразование информации из одного структурированного типа в другой может в ряде случаев происходить без потерь, либо с потерей незначительного числа информации, избыточной для конечного применения. Причем такие преобразования можно сделать с минимальными затратами времени. Так, например, информация от серверов Z39.50 к WWW-шлюзам может быть передана с помощью GRS-1, или SQL-RS, а там уже переработана в XML, HTML, GPEG и т.д.

Поддержка различных реляционных СУБД

Поскольку роль Z39.50 в связке с реляционными СУБД заключается именно в универсальности протоколов общения и схем работы, то естественным образом возникает вопрос о поддержке различных СУБД серверами Z39.50. В ZooPARK функцию взаимодействия с серверами баз данных выполняют специальные модули – провайдеры данных. На уровне провайдера полностью скрываются различия конкретных СУБД. Для остальных модулей сервера все источники данных равноправны. Таким образом, для каждой СУБД создается свой провайдер. Логика работы провайдеров полностью соответствует логике Z39.50 (search, present, scan и т.д.). Логика же работы с реляционными СУБД, как было отмечено выше, другая. Поэтому имеет смысл выделить модули для работы с реляционными СУБД в отдельную группу и проработать для нее логику работы, схожую с логикой взаимодействия с реляционными СУБД (connect, select_db, query, return_meta, return_data, close и т.д.). В настоящее время нами прорабатывается этот вариант. Такой подход позволит избавиться от многократного дублирования сложной логики провайдеров, облегчит поддержку различных реляционных СУБД, позволит вынести общую часть таких провайдеров (например, преобразование запросов, форматирование записей) в отдельный блок. Более того, при таком подходе появится возможность взаимодействовать с различными реляционными СУБД гибче, чем на уровне search- и present-запросов Z39.50.

Заключение

Несмотря на перечисленные сложности, возникающие при создании приложений, осуществляющих доступ к реляционным данным через Z39.50, информационные системы, построенные по этому принципу, могут сочетать в себе присущую Z39.50 универсальность при обмене данными и мощь реляционных СУБД при хранении и обработке информации. Надеемся, что данный материал будет интересен как разработчикам

приложений Z39.50, использующих реляционные СУБД, так и широкому кругу специалистов в области открытых информационных систем.

Литература

1. ANSI/NISO Z39.50-1995. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. Z39.50 Maintenance Agency Official Text for Z39.50-1995, July 1995 (http://lcweb.loc.gov/z3950/agency).
2. *Жижимов О.Л.* Введение в Z39.50. Новосибирск: Изд-во НГОНБ, 2000 (http://geolibr.uiggm.nsc.ru/docs/Z39.50/Z-Intro).
3. *Жижимов О.Л., Мазов Н.А.* Модель распределенной информационной системы Сибирского Отделения РАН на базе протокола Z39.50. Электронные библиотеки, 1999, т.2, вып.2.
4. Z+SQL Profile. Final as of February 23, 2000 (http://lcweb.loc.gov/z3950/agency).
5. *Гроff Д.Р., Вайнберг П.Н.* SQL:Полное руководство. Киев: BHV, 1998.

THE ACCESS TO RELATIONAL DATABASE MANAGEMENT SYSTEMS VIA Z39.50 PROTOCOL

Oleg Zhizhimov (zhizhim@uiggm.nsc.ru), *Sergey Skibin* (skibin@uiggm.nsc.ru)

UIGGM SB RAS

Koptyug Avenue 3

Novosibirsk 630090 Russia

The subject of the report is a model of “Z39.50-server + relational DBMS” system: the specificity of requirement processing and reduction of relational data to Z39.50 format. Some complexities appearing in the process of conversion of relational data to hierarchical Z39.50 data schema are examined. Different representations of structured Z39.50 data (GRS-1, SQL-RS, XML) are analyzed. The report is based on the authors’ experience obtained while Z39.50-server named ZooPARK is worked out. The prototype of system is available at http://z3950.uiggm.nsc.ru/zgwk/kadrs.htm.