ПРОГРАММНЫЕ ИНТЕРФЕЙСЫ ДЛЯ УПРАВЛЕНИЯ ДАННЫМИ, ОПИСЫВАЮЩИМИ ОБЪЕКТЫ С ИЕРАРХИЧЕСКОЙ СТРУКТУРОЙ

М. В. Конин, О. Д. Соколова
Институт Вычислительной математики и математической геофизики СО РАН

УДК 519.173, 004.42

Описываются программные интерфейсы, разработанные для управления объектами многоуровневых иерархических структур - хранения, извлечения и анализа данных. Предложены различные способы управления данными, различающиеся использованием форматов данных.

Ключевые слова: графы, гиперсети, базы данных, программный интерфейс.

Введение. Структура современных коммуникационных сетей (как транспортных сетей, так и сетей передачи информации) имеет иерархический характер. Для адекватного отображения процессов функционирования таких сетей (например, движение автомобильных потоков, передача информации) используются сложные математические объекты - модели на базе случайных и нечётких графов, гиперграфов, многоуровневые сети, в частности, гиперсети. С помощью аппарата гиперсетей [1] возможно адекватное описание таких сложных систем, как структурированные кабельные системы, транспортные сети, инженерные коммуникации. Применение гиперсетевых моделей при решении различных оптимизационных задач (управление транспортными потоками, мониторинг передачи данных и др.) описано в [2-5]. В работе [6] рассматриваются методы хранения информации о гиперсетях в документо-ориентированных базах данных.

При решении прикладных задач анализа функционирования многоуровневых сетей приходится моделировать отдельные уровни сетей традиционными графовыми моделями. При этом необходимо разрабатывать алгоритмы, которые будут учитывать иерархию уровней - сложные взаимосвязи между вершинами разных уровней, вложение элементов графа одного уровня в элементы другого и т.д.

Для удобства использования моделей с иерархической структурой необходимо наличие удобного языка - как для описания входных данных, так и для алгоритмов решения оптимизационных задач и описания полученного результата. В статье [6] были предложены некоторые способы описания элементов гиперсетей, удобные для использования их в программном интерфейсе, для хранения этих данных. В статье [7] описан разработанный автором программный комплекс Hyperman (управление объектами гиперсети)**,** который решает задачи хранения, извлечения и анализа данных для применения их в решении задач с объектами многоуровневой структуры. Здесь далее предложены программные интерфейсы для удобного пользования системой Hyperman.

1. Модели для анализа иерархических структур. Будем использовать определения из теории графов и теории гиперсетей [8].

Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проекты № 17-07-00775 и № 17-47-540977 р_а).

Определение. Гиперсетью называется шестерка $H=(X; V; R; P; F; W)$, состоящая из следующих объектов:

$X= (x_1, x_2 \ldots, x_n)$ – множество вершин;

$V= (v_1, v_2, \ldots, v_g)$ – множество ветвей;

$R= (r_1, r_2 \ldots, r_m)$ – множество ребер;

P – отображение, сопоставляющее каждому элементу v из V множество $P(v)$ его вершин, т.е. отображение P определяет гиперграф первичной сети $PH=(X,V; P)$;

F – отображение, сопоставляющее каждому элементу r из R множество $F(r)$ его ветвей. Отображение F определяет гиперграф $FH= (V; R; F)$;

W – отображение, сопоставляющее каждому элементу r из R множество $W(r)$ его вершин. Отображение W гиперграф вторичной сети $WH = (X, R; W)$.

Если для каждой ветви v выполнено условие $|P(v)|=2$, для каждого ребра r выполнено условие $|W(r)|=2$, то первичная и вторичная сети гиперсети H являются графами. В этом случае H является простой гиперсетью.

В более общем случае, когда первичная и вторичная сети являются гиперграфами, используется понятие S-гиперсети [8].

Одним из примеров сложной системы, для описания которой удобно использовать математический объект S-гиперсеть, является транспортная система мегаполиса. Всю транспортную инфраструктуру города можно рассматривать как взаимосвязанную систему сетей различного вида. При этом математическая модель, описывающая такую сложную систему, должна объединять все факторы, влияющие на ее функционирование. Например, вершинами различных графов, составляющих уровни в иерархической системе транспортной сети, могут быть как узлы уровня дорожного движения (перекрестки дорог, стоянки, автостанции), так и средства управления движением (светофоры, дорожные знаки и указатели, посты ДПС). Различные уровни транспортной сети задают структуру S-гиперсети, которую можно использовать для формального описания транспортных потоков, для решения задачи оптимального управления движением и т.д. С помощью такой модели можно описывать передвижение пассажирских и грузовых потоков; решать задачи управления потоками путем расстановки разметок и развязок; оптимизировать количество и места установки заправок, станций техобслуживания и т.д.

2. Используемые программные средства для работы с объектами иерархической структуры. Для хранения данных об элементах многоуровневой структуры была разработана система Hyperman [7]. Программный интерфейс реализован на основе фреймворка Phoenix, язык программирования Elixir. Elixir является функциональным, распределенным языком программирования общего назначения, работает на виртуальной машине Erlang (BEAM). Такой выбор позволяет разработанной системе использовать режим мягкого реального времени и горизонтально масштабироваться при увеличении вычислительных нагрузок.

2.1. *Спецификация JSON API.* При построении API (программный интерфейс приложения, интерфейс прикладного программирования, application programming interface, API), для обеспечения легкости реализации клиентских приложений и оптимизации расходов на передачу информации, использовалась спецификация JSON API. Данная спецификация декларирует, как должны выглядеть ответы и запросы к серверу от сторонних программ по протоколу HTTP. Спецификация также определяет метод работы со связями объектов, что необхо-

димо при описании многоуровневых структур. JSON API удобен для минимизации количества запросов и количества данных, передаваемых между клиентами и серверами.

Основой JSON API является ресурс, т.е. некоторый объект с определенным количеством атрибутов и связей, который поддерживает один или несколько из следующих методов: Create (создание ресурсов), Index (получения списка ресурсов), Show (возвращает ресурс по уникальному идентификатору), Update (обновляет данные ресурса), Destroy (удаляет ресурс по уникальному идентификатору).

На основе спецификации JSON API были реализованы шесть ресурсов, которые предоставляют возможность манипулировать данными об иерархических структурах, в частности, об объекте гиперсеть: гиперсеть, граф, вершина, ребро, отображение ребер, отображение вершин.

2.2. *Стандарт GraphQL*. GraphQL — это стандарт декларирования структуры данных и способов получения данных с сервера. В отличии от работы со спецификацией JSON API, клиент, который взаимодействует с сервером, декларирует те данные, которые он хочет получить. Если необходимо получить только идентификаторы всех вершин, то через JSON API сделать это нет возможности, будут получены все данные обо всех вершинах, что в некоторых случаях является избыточным. При работе со стандартом GraphQL пользователь в запросе может указать необходимые данные и сервер должен будет под это подстроиться.

В стандарте GraphQL есть два полностью отделенных друг от друга понятия: тип запроса (формат данных о нем) и способ его получения. В отличие от JSON API, GraphQL дает пользователю более удобные возможности по выбору данных. При формировании запроса пользователь вправе сам решать, какие именно данные и в каком виде ему необходимы, т.е. возникают преимущества, которые можно использовать при проектировании алгоритмов решения прикладных задач.

3. Разработанные программные средства хранения объекта гиперсеть и извлечения необходимых данных. В работе [7] приводится описание разработанного программного комплекса, который решает задачи хранения, извлечения и анализа данных для многоуровневых структур.

Предложенное решение состоит из следующих компонент:

– документно-ориентированная база данных для хранения информации (MongoDB);

– веб-приложение для извлечения данных из базы и преобразования их в приведенные выше форматы;

– балансировщик нагрузок для распределения запросов.

Для сравнения разработанного метода хранения данных с существующими методами, описывающими сложные иерархические структуры, рассмотрим модель, предложенную в работе [9]. Авторы описывают многоуровневую модель, где данные хранятся в виде древовидной (иерархической) структуры, состоящей из объектов различных уровней. Между объектами существуют связи, каждый объект может включать в себя несколько или ни одного объектов более низкого уровня. Такая модель является хорошей основой для анализа сложных структур, но не может быть применена для всех возможных типов иерархических объектов, а только лишь для некоторых конкретных случаев. Например, в ней нельзя декларировать типы самих связей, что является необходимым в задачах оптимизации транспортных сетей.

Для создания более адекватного способа хранения и извлечения данных иерархических структур предлагается использовать реализации приведенных выше спецификаций JSON API и GraphQL. Кроме низкоуровневых API, для работы с многоуровневыми объектами разработан предметно-ориентированный язык DSL - Domain-specific language. DSL является расширением языка программирования общего назначения ruby. Следовательно, он содержит все базовые конструкции ruby, а именно: условия, циклы, обработчики ошибок и т.д. Кроме этого, язык ruby имеет стандартную библиотеку, которая содержит много полезных функций, включая удобную работу с файлами. При объединении с представленным DSL, это дает большие возможности экспорта данных из веб-приложения в нужные форматы, что необходимо при использовании различных алгоритмов.

Разработанный язык базируется на семи основных классах, каждый из которых отвечает за определенные операции:

1. Hypernet – операции с гиперсетями
2. Graph – операции с графами
3. Node – операции с вершинами
4. Edge – операции с ребрами
5. Mapping – операции с отображениями
6. NodeMapping – операции с отображениями вершин
7. EdgesMapping – операции с отображениеми ребер

Каждый из этих классов имеет следующие методы:

– метод создания экземпляра с сохранением на сервере (create);

–метод поиска в базе данных (query);

– метод поиска объекта по уникальному идентификатору (find);

– метод удаления (destroy);

– метод обновления данных (update);

– метод сохранения данных на сервер (save).

Каждый объект перечисленных классов содержит все атрибуты и все связи, доступные через JSON API.

Данный язык разрабатывался главным образом для того, чтобы дать пользователю человеко-понятный и выразительный способ описания моделей многоуровневых структур. Также, одной из целей была инкапсуляция сетевого взаимодействия. Для того, чтобы пользоваться приведенными выше стандартами JSON API и GraphQL, необходимы определенные познания в области веб-технологий, т.е. пользователь должен будет использовать некоторую библиотеку для взаимодействий по сети. Разработанный предметно-ориентированный язык DSL инкапсулирует все сетевые взаимодействия и дает пользователю только определенный набор объектов и функций, необходимых для анализа многоуровневых систем.

Заключение. Разработанные способы взаимодействия пользователя с web-системой управления хранением объектов иерархических структур дают пользователю возможность работать с многоуровневыми моделями и решать прикладные задачи с учетом межуровневых связей в сложных объектах. Это актуально для решения задач анализа функционирования современных сложных систем, моделируемых иерархическими структурами - транспортных сетей, сетей передачи данных и т.д.

Список литературы

1. Попков В.К. Математические модели связности. М: Издательство ИВМиМГ СО РАН, Новосибирск. 2006. – 490 стр.

2. Rodionov A., Rodionova O. Random hypernets in reliability analysis of multilayer networks // Lecture Notes in Computer Science. 2015. Vol. 343. P. 307-315.

3. Попков В.К. О моделировании городских транспортных систем гиперсетями. Автоматика и телемеханика, 2011. Вып. 6. С. 179–189.

4. Попков Г. В. Программно-алгоритмическое обеспечение для реализации нового подхода к построению единой транспортной сети связи // Программные продукты и системы. №4. 2014. С. 242-246.

5. Соколова О.Д., Юргенсон А.Н. Задача оптимального размещения устройств анализа информационных потоков в сетях // Проблемы информатики. 2010. № 2. С. 33-42.

6. Конин М. В. Методы хранении и анализа гиперсетей в документо-ориентированных базах данных // Труды 12-й Международной Азиатской школы-семинара «Проблемы оптимизации сложных систем», Новосибирск, 12-16 декабря 2016 г. С. 285-289.

7. Конин М. В. Представление данных для иерархических структур в документо-ориентированных базах данных // Вестник СибГУТИ, 2018/ (принято в печать).

8. Попков В. К. Применение теории S-гиперсетей для моделирования систем сетевой структуры. // Проблемы информатики. 2010. С. 17-40.

9. Silberschatz A. et al. Database system concepts. – New York: McGraw-Hill– T. 4. 1997.

Конин Максим Васильевич – асп. Института вычислительной математики и математической геофизики СО РАН;
630090, Новосибирск; e-mail: maxim21214@gmail.com;
Соколова Ольга Дмитриевна – канд. техн. наук, ст. науч. сотр. Института вычислительной математики и математической геофизики СО РАН; 630090, Новосибирск; e-mail: olga@rav.sscc.ru