ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Գրիգոր Ռուբենի Գևորգյան

ՏՎՅԱԼՆԵՐԻ ԻՆՏԵԳՐՄԱՆ ՄԻ ՄՈՏԵՑՄԱՆ ՄԱՍԻՆ՝ ՄՈԴԵԼ, ԱԼԳՈՐԻԹՄՆԵՐ ԵՎ ՎԵՐԻՖԻԿԱՑԻԱ

ՍԵՂՄԱԳԻՐ

Ե13.04 – «Հաշվողական մեքենաների, համալիրների, համակարգերի և ցանցերի մաթեմատիկական և ծրագրային ապահովում» մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման ատենախոսության

Երևան – 2016

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА

Григор Рубенович Геворгян

ОБ ОДНОМ ПОДХОДЕ К ИНТЕГРАЦИИ ДАННЫХ: МОДЕЛЬ, АЛГОРИТМЫ И ВЕРИФИКАЦИЯ

АВТОРЕФЕРАТ

Диссертации на соискание ученой степени кандидата технических наук по специальности 05.13.04 – “Математическое и программное обеспечение вычислительных машин, комплексов, систем и сетей”

Ереван – 2016

Ատենախոսության թեման հաստատվել է Ռուս-Հայկական (Սլավոնական) համալսարանում

Գիտական ղեկավար՝ ֆ.մ.գ.դ. Ս. Կ. Շուքուրյան
Պաշտոնական ընդդիմախոսներ՝ ֆ.մ.գ.դ. Լ. Հ. Ասլանյան
ֆ.մ.գ.թ. Ռ. Վ. Թոփչյան

Առաջատար կազմակերպություն՝ Հայաստանի ազգային պոլիտեխնիկական համալսարան

Պաշտպանությունը կայանալու է 2016թ. Հունիսի 13-ին, ժ. 16:00-ին ՀՀ ԳԱԱ Ինֆորմատիակայի և ավտոմատացման պրոբլեմների ինստիտուտում, թիվ 037 «Ինֆորմատիկա և հաշվողական համակարգեր» մասնագիտական խորհրդի նիստում, հետևյալ հասցեով՝ 0014, Երևան, Պ. Սևակի 1

Ատենախոսությանը կարելի է ծանոթանալ ՀՀ ԳԱԱ ԻԱՊԻ-ի գրադարանում:

Սեղմագիրն առաքված է 2016թ. Մայիսի 13-ին:

037 մասնագիտական խորհրդի գիտական քարտուղար, ֆ.մ.գ.դ. Հ. Գ. Սարուխանյան

Тема диссертации утверждена в Российско-Армянском (Славянском) университете

Научный руководитель: д.ф.м.н. С. К. Шукурян
Оффициальные оппоненты: д.ф.м.н. Л. А. Асланян
к.ф.м.н. Р. В. Топчян

Ведущая организация: Национальный политехнический университет Армении

Защита состоится 13-го июня 2016г. в 16:00 на заседании специализированного совета 037 “Информатика и вычислительные системы” в Институте проблем информатики и автоматизации НАН РА по адресу: 0014, г. Ереван, ул. П. Севака 1.

С диссертацией можно ознакомится в библиотеке ИПИА НАН РА.

Автореферат разослан 13 мая 2016г.

Ученый секретарь специализированного совета 037, д.ф.м.н. А. Г. Саруханян

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность проблемы

Целью интеграции данных является использование содержимого двух или более источников данных (баз данных) в рамках более крупной, возможно виртуальной, базы данных с целью обращения к ней с запросами как к унифицированному информационному пространству. Интеграционная система должна предоставлять пользователю унифицированный взгляд на множество неоднородных источников информации, называемый *глобальной схемой*.

Основными технологиями, используемыми при решении задачи интеграции данных, являются *федеративные базы данных*, *медиаторы* и *хранилища данных*. При использовании федеративных баз данных источники информации независимы, но каждый из них способен получать требуемую информацию из других. Примерами современных приложений, использующих технологии федеративных баз данных, являются IBM DB2, Oracle Data Integrator и GoldenGate.

Медиаторы представляют из себя программные компоненты, обеспечивающие поддержку виртуальных баз данных (virtual databases), которые “внешне” выглядят так, словно они материализованы (т.е. сконструированы физически, подобно хранилищам данных). Медиатор сам по себе не сохраняет информацию – вместо этого он транслирует запрос пользователя в один или несколько запросов, адресованных первичным источникам. Получая результаты обработки частных запросов, медиатор синтезирует на их основе ответ на исходный запрос. Системы интеграции данных формально определяются в виде тройки $\langle G, S, M \rangle$, где G – глобальная схема (схема медиатора), S – множество схем источников, а M – отображение между запросами над глобальной схемой и схемами источников. В архитектурах, основанных на идее медиатора, именуемых также архитектурами с посредником, используются три разновидности представления интегрированных данных – Global as View (GAV), Local as View (LAV) а также их совмещение, именуемое GLAV. Согласно GAV, глобальная схема определяется как взгляд над источниками данных, в то время как GAV предполагает, что источники данных определены как взгляд над глобальной схемой. Подход GAV имеет лучшую производительность обработки запросов чем LAV, но LAV имеет лучшую расширяемость чем GAV. Для совмещения преимуществ упомянутых технологий был предложен смешанный подход[1], называемый GLAV. Существует метод компиляции системы GLAV в эквивалентную ей систему GAV[2]. Подход к интеграции данных, предлагаемый в данной работе, относится к разновидности GLAV. Среди современных исследований в области виртуальной интеграции

[1] M. Friedman, A. Y. Levy and T. D. Millstein, "Navigational plans for data integration," in Sixteenth National Conference on Artificial Intelligence and Eleventh Conference, Florida, USA, 1999

[2] A. Cali, "Reasoning in data integration systems: why lav and gav are siblings," in ISMIS, Maebashi City, Japan, 2003.

данных следует выделить работы группы SYNTHESIS[3,4], которые являются пионерами в области исследования методов интеграции неоднородных источников информации. Группой SYNTHESIS были предложены концепция канонической модели данных и принцип коммутативных отображений моделей данных[5], который является основой предлагаемого в данной работе подхода к интеграции данных. Также следует выделить исследования Паоло Атцени[6,7], которые ориентированы на задачи интеграции как структурированных, так и NoSQL баз данных.

При использовании хранилищ данных копии фрагментов информации из нескольких источников сохраняются в единой базе данных, возможно, с предварительной обработкой – фильтрацией (filtering), соединением (joining) или агрегированием (aggregating). Содержимое хранилища обновляется на периодической основе. В процессе копирования данные подвергаются определенным преобразованиям с целью согласования их структур с общей схемой хранилища. Основным недостатком хранилищ данных является их потенциальное несоответствие реальному времени, так как изменения в источниках данных могут не быть своевременно в них отражены[8]. Важными классами приложений систем интеграции информации, основанных на технологии хранилищ данных, на сегодняшний день являются приложения *OLAP* (от on-line analytic processing – оперативная аналитическая обработка данных) и приложения *интеллектуального анализа данных* (data mining).

Возникновение новой парадигмы в науке и различных приложениях информационных технологий связано с проблемами обработки *больших данных* (big data). Концепция больших данных относительно нова, и следующие пять характеристик обычно используются в качестве определяющих: объем (volume), скорость (velocity), многообразие (variety), достоверность (veracity) и значимость (value)[9]. Одной из ключевых задач в данной области является интеграция неоднородных источников информации.

Цель работы

Целью работы является исследование задач интеграции данных, разработка подхода к интеграции данных на основе принципа коммутативных отображений моделей данных,

---

[3] http://synthesis.ipi.ac.ru

[4] С. Ступников, Н. Скворцов, В. Буздко, В. Захаров, Л. Калиниченко, "Методы унификации нетрадиционных моделей данных", *Системы высокой доступности*, с. 18-39, Москва, Радиотехника, 2014

[5] Л. Калиниченко, С. Ступников и Н. Земцов, "Методы синтеза канонических моделей, предназначенных для достижения семантической интероперабельности неоднородных источников информации", ИПИ РАН, Москва, 2005

[6] L. Bellomarini, P. Atzeni and L. Cabibbo, "Data integration with many heterogeneous sources and dynamic target schemas (extended abstract)", *in Proceedings of AMW2015*, pp. 148-155, Lima, Peru, 2015

[7] P. Atzeni, L. Bellomarini and F. Bugiotti, "Exlengine: Executable schema mappings for statistical data processing" *in Proceedings of EDBT'13*, pp. 672-682, New York, NY, USA, 2013

[8] H. Garcia-Molina, J. Ullman and J. Widom, "Database Systems: The Complete Book", Prentice Hall, 2009.

[9] S. Sharma, U. S. Tim, J. Wong, S. Gadia and S. Sharma, "A Brief Review on Leading Big Data Models" in *Data Science Journal*, 2014.

включающего в себя разработку расширяемой канонической модели и метода верификации корректности отображений моделей данных, а также алгоритмов поддержки виртуальной и материализованной интеграции.

Методы исследования

При решении поставленных в работе задач использовались методы Нотации Абстрактных Машин (Abstract Machine Notation – AMN)[10], теории множеств и логики предикатов.

Научная новизна

В диссертационной работе получены следующие результаты:

1. Разработан подход к интеграции данных на основе принципа коммутативных отображений моделей данных;
2. Предложена расширяемая XML-каноническая модель данных;
3. Построены AMN-машины для канонической и реляционной моделей данных и их обратимого отображения и доказана его корректность;
4. Разработан подход к построению хранилища данных, основанный на динамической структуре индекса для многомерных данных, предложены эффективные алгоритмы для ее поддержки и приведены оценки сложности предложенных алгоритмов;
5. Разработан и реализован прототип хранилища данных на основе предложенной динамической структуры индекса.

Практическое значение

Предлагаемый в работе подход к интеграции данных может быть использован для интеграции неоднородных источников информации в приложениях аналитической обработки в режиме реального времени, приложениях интеллектуального анализа данных, а также в области больших данных. Эксперименты, проведенные с использованием разработанного прототипа хранилища, показывают, что использование предложенного подхода может привести к значительному уменьшению размера директории индекса и времени поиска для многомерных данных.

Положения, выносимые на защиту

1. Расширяемая XML-каноническая модель данных;
2. Подход к виртуальной интеграции данных;
3. Подход к материализованной интеграции данных;
4. Прототип хранилища данных на основе предложенного подхода к материализованной интеграции данных.

[10] J.-R. Abrial, The B-Book - Assigning programs to meaning, Cambridge University Press, 1996.

Апробация работы

Основные результаты диссертационной работы обсуждались на семинарах кафедры системного программирования Российско-Армянского (Славянского) Университета (РАУ) и кафедры информационных систем Образовательного и исследовательского центра информационных технологий Ереванского Государственного Университета (ЕГУ). Основные результаты диссертационной работы докладывались на симпозиуме First Workshop on Programming the Semantic Web в рамках конференции International Semantic Web Conference 2012г, а также на годичных научных конференциях РАУ 2011, 2012, 2015гг.

Публикации

Основные результаты исследований отражены в 6 научных публикациях [1-6].

Структура и объем работы

Диссертация состоит из введения, четырех глав, заключения и списка использованной литературы (92 наименования). Общий объем диссертации – 102 страницы.

Благодарности

Автор выражает глубокую благодарность к.ф.м.н., доценту М. Г. Манукяну за неоценимые советы, поддержку и помощь в работе на протяжении всех этапов исследования.

СОДЕРЖАНИЕ ДИССЕРТАЦИИ

Во введении обоснована актуальность и практическая значимость темы диссертационной работы, кратко изложено состояние предметной области, сформулированы цели и основные задачи исследования, выделены научные результаты, отличающиеся новизной, выносимые на защиту научные положения и практическая ценность полученных результатов.

В первой главе диссертации был проведен анализ основных подходов к интеграции данных. Рассмотрены основные технологии интеграции данных – федеративные базы данных, хранилища данных и медиаторы. Сформулированы задачи, возникающие при виртуальной и материализованной интеграции данных.

В §1.1 описана технология федеративных баз данных, рассмотрены примеры использующих ее современных приложений.

В §1.2 приведено описание технологии хранилищ данных. Рассмотрена система управления данными с открытым кодом SciDB [11], нацеленная преимущественно на использование в областях, в которые вовлечены массивы данных очень больших размеров.

В §1.3 рассмотрена технология медиаторов. Описаны три разновидности представления интегрированных данных, применяемых в архитектурах, основанных на идее медиатора

[11] http://paradigm4.com

(архитектурах с посредником) – Global as View, Local as View а также их совмещение, именуемое GLAV. Приведен обзор исследований Паоло Атцени, описана архитектура системы SOS[12].

Особое внимание уделено подходу к интеграции данных, предложенному группой SYNTHESIS. В работах группы SYNTHESIS были впервые определены понятия эквивалентности состояний баз данных, схем и моделей данных и описаны принципы сохранения операций и информации в процессе конструирования отображений неоднородных источников информации в целевую (каноническую) модель. В основе этих принципов лежит идея расширения канонической модели данных таким образом, чтобы стало возможным отображение информации и операторов в нее из моделей данных источников. Все операции над схемами и моделями данных производятся на уровне некоторой метамодели данных – формального языка, обладающего достаточной мощью, чтобы иметь возможность представлять всевозможные концепции различных моделей данных. В качестве такой метамодели данных используется нотация абстрактных машин. В рамках этой нотации понятие эквивалентности моделей данных сводится к понятию уточнения соответствующих спецификаций. Основным преимуществом использования AMN является B-технология, предоставляющая возможность полуавтоматического доказательства факта уточнения. Наличие подобной возможности доказательства существования отображений из исходных моделей данных в целевую является причиной выбора данного подхода в качестве основы для предлагаемого в диссертационной работе подхода к интеграции данных.

Согласно подходу группы SYNTHESIS, в рамках СУБД каждая модель данных определяется синтаксисом и семантикой двух языков – языка определения данных (ЯОД) и языка манипулирования данными (ЯМД). Основными принципами отображения произвольной исходной модели данных в целевую являются[13]:

1. *Принцип аксиоматического расширения моделей данных*. Согласно этому принципу, каноническая модель должна быть расширяемой. При этом ее расширение при рассмотрении каждой новой модели данных носит аксиоматический характер: целевая модель данных расширяется путем добавления к ее ЯОД набора аксиом, определяющих логические зависимости данных исходной модели в терминах целевой. Полученное расширение должно быть эквивалентно исходной модели.
2. *Принцип коммутативного отображения моделей данных*. Согласно этому принципу, сохранение операций и информации исходной модели при ее отображении в каноническую достигается при условии коммутативности диаграммы отображения ЯОД (схем) и ЯМД (операций).

---

[12] P. Atzeni, F. Bugiotti и L. Rossi, “Uniform access to non-relational database systems: The SOS platform”, Advanced Information Systems Engineering, pp. 160-174, 2012

[13] Л. Калиниченко, С. Ступников и Н. Земцов, “Методы синтеза канонических моделей, предназначенных для достижения семантической интероперабельности неоднородных источников информации”, ИПИ РАН, Москва, 2005

Множество всех схем, выразимых в ЯОД модели данных M_i, обозначается S_i, а множество операторов ЯМД модели M_i обозначается O_i . *Пространство допустимых состояний*, выразимых в M_i, обозначается B_i. Тогда

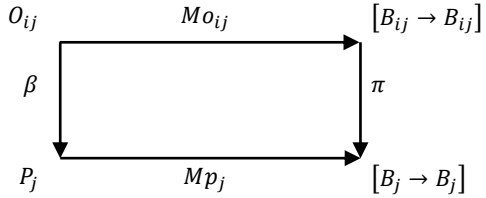
$Ms_i : S_i \to B_i$ есть семантическая функция ЯОД M_i.

$Mo_i : O_i \to [B_i \to B_i]$ есть семантическая функция ЯМД M_i.

При этих обозначениях, отображение $f = \langle \sigma, \theta, \beta \rangle$ модели M_j в расширение M_{ij} модели M_i коммутативно, если выполняются следующие условия:

- Диаграмма отображения схем является коммутативной:

- Диаграмма отображения операторов является коммутативной:



- Отображение θ биективно.

Здесь Ω_{ij} обозначает множество схем аксиом, выражающих зависимости данных модели M_j в терминах модели M_i, а P_j обозначает последовательность операторов ЯМД модели M_j.

3. *Принцип синтеза унифицирующей канонической модели данных*. Синтезом канонической модели называется процесс построения расширений его ядра, эквивалентных различным моделям данных, включаемых в среду, а также процесс слияния этих расширений с канонической моделью. Согласно этому принципу, в создаваемой унифицирующей канонической модели разнообразные исходные модели данных имеют однородное эквивалентное представление.

Далее следует описание нотации абстрактных машин, которая была впервые применена в качестве формальной метамодели данных в начале 90-х годов. AMN обеспечивает манипулирование теоретико-множественными спецификациями в логике первого порядка и доказательство уточнения спецификаций. Техника уточнения позволила расширить основные определения отношений между типами данных, схемами, моделями данных так, чтобы вместо эквивалентности соответствующих спецификаций можно было рассуждать об их уточнении. Специальные инструментальные средства (B-технология) представляют возможность доказательства коммутативности диаграмм отображения моделей полуавтоматической

способом: теоремы, требуемые для доказательства уточнения моделей, генерируются B автоматически, но их доказательство может требовать вмешательства человека.

Вторая глава посвящена созданию расширяемой канонической модели в рамках предлагаемого подхода к интеграции данных. Предложен принцип расширения ядра канонической модели. В целях создания обоснованного отображения для интеграции неоднородных баз данных, концепция модели данных формализуется посредством нотации абстрактных машин. Для каждой исходной модели создается обратимое отображение в расширение канонической модели. B-технология используется для доказательства того, что AMN-семантика исходной модели представляет собой AMN-семантику расширенной канонической модели. Таким образом доказывается корректность отображения и возможность использования расширенной канонической модели для представления схем исходной модели. Созданы AMN-машины для канонической и реляционной моделей, а также для реляционного уточнения канонической модели.

В §2.1 приведено описание модели данных XDM[14], которая была выбрана в качестве ядра канонической модели данных для медиатора. Выбор обусловлен рядом преимуществ этой модели данных. В частности, она представляет из себя компромисс между структурированными (реляционная, объектная) и полуструктурированными (XML) моделями данных. XDM появилась в результате расширения модели данных XML в целях поддержки концепции базы данных. В рамках этой модели рассматриваются базовые объекты, такие как целые числа, строки, переменные различных типов, символы, и составные объекты, определенные в терминах λ-исчисления.

В §2.2 предложен подход к интеграции данных, основанный на онтологии. В рамках данного подхода рассматривается как виртуальная, так и материализованная интеграция данных в пределах канонической модели. Таким образом, возникает необходимость формализации понятий данной предметной области, таких как медиатор, хранилище данных и схема базы данных. В данном случае онтология интеграции данных основана на XML формализации этих понятий.

В §2.3 сформулирован принцип расширения ядра канонической модели. Расширение ядра канонической модели производится при рассмотрении моделей данных новых источников информации путем добавления новых символов (понятий) к ее ЯОД в целях выражения логических зависимостей исходной модели средствами целевой. Полученное расширение должно стать эквивалентным исходной модели данных. Для реализации расширения канонической модели используется следующее правило:

Concept ← *Symbol* ContextDefinition

Например, для поддержки таких понятий реляционной модели как ключ и ссылочная целостность, ядро канонической модели дополняется следующими символами: *key, foreign key,*

[14] M. G. Manukyan, “Extensible Data Model”, in Advances in Databases and Information Systems, 2008

unique, constraint, on update, on delete, cascade, set null. В общем случае, расширение ядра сводится к добавлению новых символов в так называемые *словари контента* (content dictionaries), представленные в виде XML документов, и определению контекста применения данных символов. Существенно, что для определения контекста используется вычислительно полный язык[15]. Словари контента используются для присвоения формальной и неформальной семантики всем понятиям, используемым в моделях данных. В результате данного подхода использование новых символов в ЯОД не приводит к изменению парсера ЯОД.

В §2.4 предложен метод формализации моделей данных. В рамках предлагаемого подхода к интеграции данных, модели данных и отображения между ними рассматриваются как экземпляры метамодели. В качестве метамодели используется нотация абстрактных машин. Иными словами, AMN-формализм используется для определения концепций моделей данных. Каждая используемая модель данных формализуется в AMN в виде некоторой абстрактной машины (либо иерархии абстрактных машин), представляющей AMN-семантику для этой модели. Абстрактная машина Mch_M для модели данных M должна быть построена таким образом, чтобы множество схем модели M находилось в биекции с множеством допустимых состояний машины Mch_M, ограниченным ее инвариантом I_M. Отображение из исходной модели в целевую сводится к моделированию понятий целевой модели посредством понятий исходной модели. Для этой цели строится AMN-машина, являющаяся расширением AMN-машины исходной модели и уточнением AMN-машины целевой модели данных.

Предложены следующие принципы AMN-формализации моделей данных:

1. Базовые понятия моделей данных формализуются с помощью системы типов AMN;
2. Сложные понятия моделей данных формализуются с помощью абстрактных машин.

Пусть построены абстрактные машины Mch_T для канонической модели и Mch_S для модели данных источника. Каноническая модель расширяется с помощью машины Ext_{ST}, содержащей концепции исходной модели, отсутствующие в целевой. Эти понятия представлены в виде множеств и констант машины Ext_{ST}. Ниже приведены AMN-представления исходной и канонической моделей:

MACHINE Mch_T	MACHINE Mch_S
SETS $Sets_T$	SETS $Sets_S$
CONSTANTS $Const_T$	CONSTANTS $Const_S$
PROPERTIES P_T	PROPERTIES P_S
VARIABLES Var_T	VARIABLES Var_S
INITIALISATION $Init_T$	INITIALISATION $Init_S$
INVARIANT I_T	INVARIANT I_S
OPERATIONS Op_T	OPERATIONS Op_S
END	END

[15] M. Drawar, “OpenMath: An Overview”, *ACM SIGSAM Bulletin,* vol. 34, no. 2, 2000

Следующая AMN-схема представляет собой расширение канонической модели:

MACHINE Ext_{ST}
EXTENDS Mch_T
SETS $Sets_{ST}$
CONSTANTS $Const_T$
PROPERTIES P_{ST}
END

Пусть для исходной модели M_S и канонической модели M_T уже построены соответствующие абстрактные машины. В рамках предлагаемого подхода к интеграции данных, для доказательства корректности отображения из исходной модели в каноническую, строится абстрактная машина, которая расширяет AMN-машину исходной модели и уточняет AMN-машину канонической модели. В секции инварианта данной машины определяются условия соответствия схем исходной и канонической моделей. В секции операций посредством операций модели источника моделируются операции канонической модели. Ниже представлена AMN-схема для этой машины:

REFINEMENT $Ext_{ST}Ref$
REFINES Ext_{ST}
EXTENDS Mch_S
INVARIANT I_R
OPERATIONS Op_R
END

Верификация факта уточнения производится полуавтоматически с использованием B-технологии. Таким образом доказывается корректность построенного отображения. Были построены AMN-машины для канонической и реляционной моделей данных и их обратимого отображения и доказана его корректность.

В предлагаемом подходе к интеграции информации формализм AMN имеет двойное применение. Во-первых, он используется для доказательства корректности отображения исходной модели в каноническую. Во-вторых, на основе AMN-машин исходной и канонической моделей и машины-уточнения канонической модели генерируются канонические схемы. Для поддержки канонических схем предполагается разработка словаря данных (в виде XML приложения) для канонической модели. Данный словарь данных должен содержать три типа метаданных:

1. Метаданные в контексте традиционных баз данных (данные об объектах канонической модели);
2. Метаданные для присвоения неформальной и формальной семантики всем понятиям, используемым в канонической модели;
3. Метаданные для формализации сигнатур понятий канонической модели (для проверки семантической корректности представления объектов канонической модели).

Третья глава посвящена разработке хранилища данных для предложенной канонической модели данных на основе индекса, представляющего собой расширение концепции сеточных файлов[16]. Предложенный подход к построению хранилища данных в основном ориентирован на поддержку OLAP приложений. В качестве формализма для управления многомерными данными использована концепция сеточных файлов, которая является одним из адекватных формализмов эффективного управления многомерными данными и может быть использована для эффективного хранения кубов данных в хранилищах[17]. Модель сеточного файла можно представлять так, будто пространство точек-объектов разбивается на части воображаемой сеткой. Линии сетки, параллельные осям координат, разделяют пространство на *полосы* (stripes). Количество линий сетки по различным измерениям может варьироваться, также может различаться ширина полос – даже в пределах одного измерения. Пересечения этих полос образуют прямоугольные области, именуемые *ячейками* сеточного файла. Принадлежащие ячейке точки представляются записями, хранящимися в соответствующем ей *блоке*, указатель на который в свою очередь хранится в ячейке.

На рис. 1 приведен пример 3-мерного сеточного файла. Здесь X, Y и Z являются измерениями рассматриваемого пространства, которое разделено на полосы v_1, v_2, v_3 в измерении X, w_1, w_2 в измерении Y и u_1, u_2, u_3 в измерении Z.

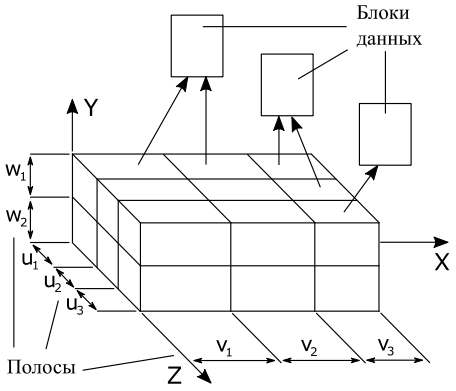


Рис. 1. Пример 3-мерного сеточного файла

Одним из недостатков, присущих сеточным файлам, является проблема неэффективного использования памяти группами ячеек, ссылающихся на одни и те же блоки данных. В данной работе предлагается альтернативная структура индекса, основанная на концепции сеточного файла и имеющая целью избежать хранения повторяющихся указателей на одни и те же блоки данных, а также поддерживать плавный рост размера директории индекса и обеспечить разумные стоимости операций.

В §3.1 предложена модификация структуры сеточного файла. В данном подходе сеточный файл не хранится в виде многомерного массива. Причина этого заключается в том, что при каждом разделении сегмента данных одна из пересекающих его полос также разделяется на две полосы, таким образом удваивая количество ячеек исходной полосы, при

[16] J. Nievergelt and H. Hinterberger, "The Grid File: An Adaptable, Symmetric, Multikey File Structure", *ACM Transactions on Database Systems*, vol. 9, no. 1, pp. 38-71, 1984

[17] C. Luo, W. C. Hou, C. F. Wang, H. Want and X. Yu, "Grid File for Efficient Data Cube Storage", *Computers and their Applications*, pp. 424-429, 2006

этом многие из новых ячеек содержат повторяющиеся указатели на одни и те же блоки данных. Вместо этого, все ячейки, для которых соответствующие записи хранятся в одном и том же блоке данных, объединяются в *сегменты* (chunks), представляемые едиными ячейками памяти с одним указателем на соответствующий блок. Сегменты являются основными единицами ввода/вывода данных, а также используются для кластеризации данных. Сегменты используются в качестве механизма разрешения проблемы пустых ячеек в сеточном файле. Для каждого измерения информация о его разделении хранится в линейной шкале, каждый элемент которой соответствует полосе сеточного файла и представляется в виде массива указателей на пересекаемые этой полосой сегменты. Каждая полоса рассматривается в качестве линейной хеш-таблицы. С целью уменьшения количества сегментов сеточного файла используются блоки переполнения. Количество блоков переполнения может различаться среди сегментов. При этом оно поддерживается на таком уровне, чтобы для каждой полосы среднее количество блоков переполнения для пересекаемых ею сегментов было меньше единицы. Это позволяет значительно уменьшить общее количество сегментов, гарантируя выполнение не более двух дисковых операций для доступа к данным.

На рис. 2 приведен пример сеточного файла, построенного с применением модифицированной концепции.

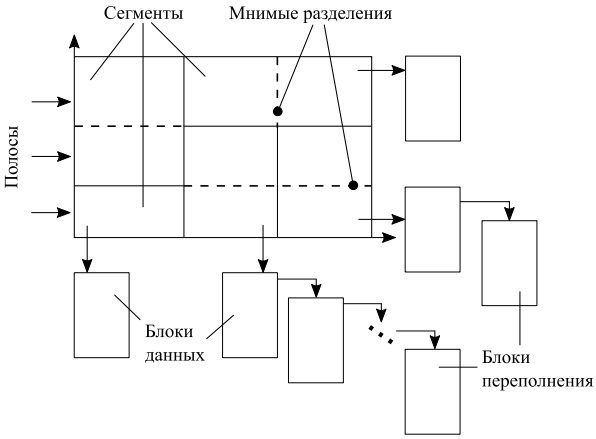


Рис. 2. Пример 2-мерного модифицированного сеточного файла

Сеточный файл формально представляется в виде тройки $F = \langle D, S, C \rangle$ где D есть множество измерений, S – множество полос, а C – множество сегментов. Каждая полоса соответствует в точности одному измерению и пересекает некоторое непустое подмножество множества сегментов. Далее рассмотрены некоторые характеристики модифицированной структуры сеточных файлов, и приведены оценки их величин. Анализ проведен при предположении, что все измерения поля данных сеточного файла являются независимыми и эквивалентными (равноправными). Количество измерений обозначается как n, а среднее количество операций разделения, осуществленных в каждом измерении, как m.

Количество ячеек в сетке. Так как каждое из n измерений разделено на m частей в среднем, общее количество ячеек в структуре сеточного файла в среднем равно m^n.

Количество полос в одном измерении. При проведении одной операции разделения может возникнуть не более одной новой полосы. Количество полос может быть уменьшено при операциях слияния. Таким образом, количество полос в одном измерении ограничено сверху

количеством операций разделения, осуществленных в данном измерении, и может быть оценено как $O(m)$.

Общее количество полос таким образом имеет порядок $O(nm)$.

Общее количество сегментов. Новые сегменты возникают только при осуществлении операций разделения, при этом в результате одной операции разделения количество сегментов увеличивается ровно на единицу. Это означает, что общее количество сегментов ограничено сверху количеством произведенных операций разделения и может быть оценено как $O(nm)$.

Среднее количество ячеек в сегменте является отношением общего количества ячеек к количеству сегментов и имеет порядок

$$O\left(\frac{m^n}{nm}\right) = O\left(\frac{m^{n-1}}{n}\right)$$

Средняя величина стороны сегмента. Для оценки в качестве единицы измерения используется средняя величина стороны ячейки сетки. Без нарушения общности можно предположить, что средней формой сегмента является *n*-мерный куб. В таком случае средняя величина его стороны будет иметь порядок

$$O\left(\sqrt[n]{\frac{m^{n-1}}{n}}\right) = O\left(\frac{m}{\sqrt[n]{nm}}\right)$$

Среднее количество сегментов, пересекаемых полосой. Полоса имеет среднюю длину m ячеек в $n-1$ измерении. Так как средняя величина стороны сегмента имеет порядок

$$O\left(\frac{m}{\sqrt[n]{nm}}\right)$$

то количество сегментов, пересекаемых полосой, будет иметь порядок

$$O\left(\left(\frac{m}{\frac{m}{\sqrt[n]{nm}}}\right)^{n-1}\right) = O\left(\left(\sqrt[n]{nm}\right)^{n-1}\right)$$

Для простоты дальнейших рассуждений мы ослабим эту оценку до $O(nm)$.

Размер директории индекса. Так как каждая из $O(nm)$ полос пересекает в среднем $O(nm)$ сегментов, общее количество хранимых указателей будет иметь порядок $O(n^2m^2)$. Также, каждый сегмент хранит один указатель на соответствующий ему блок данных. Общее количество таких указателей - $O(nm)$. Таким образом, величина директории индекса имеет порядок $O(n^2m^2)$.

Далее в §3.1 описаны операции над сеточными файлами, адаптированные к применению над предложенной модифицированной структурой. Приведены сложности операций в контексте модификации индекса и дисковых операций.

В §3.2 предложена альтернативная структура сеточного файла, являющаяся дальнейшей модификацией структуры, предложенной в предыдущем параграфе, и позволяющая уменьшить размер директории индекса от $O(n^2m^2)$ до $O(n^2m)$. Для достижения этой цели проводится реорганизация системы хранения указателей на сегменты, позволяющая сегментам хранить указатели друг на друга. Формально, множество указателей на сегменты определяется следующим образом:

Определение 3.2.1. Пусть на множестве сегментов определено отношение полного порядка $<$. Обозначим проекцию сегмента c на измерение d через $\pi_d(c)$. Множество указателей на сегменты R определяется следующим образом:

1. Для любой пары сегментов a, b, т.ч. $a < b$ и существует измерение d, т.ч. $\pi_d(a) \subseteq \pi_d(b)$, и не существует сегмента c, т.ч. $a < c < b$ и $\pi_d(a) \subseteq \pi_d(c) \subseteq \pi_d(b)$, существует указатель $(a, b) \in R$. Назовем его *указателем в измерении* d. Данный указатель хранится в списке указателей сегмента a.
2. Для любого сегмента a и полосы s измерения d, если в R не сущетвует указателя на сегмент a в измерении d, то существует указатель $(s, a) \in R$. Данный указатель хранится в списке указателей полосы s.

Пример эквивалентных сеточных файлов, построенных согласно исходной и модифицированной структурам, приведен на рис. 3.

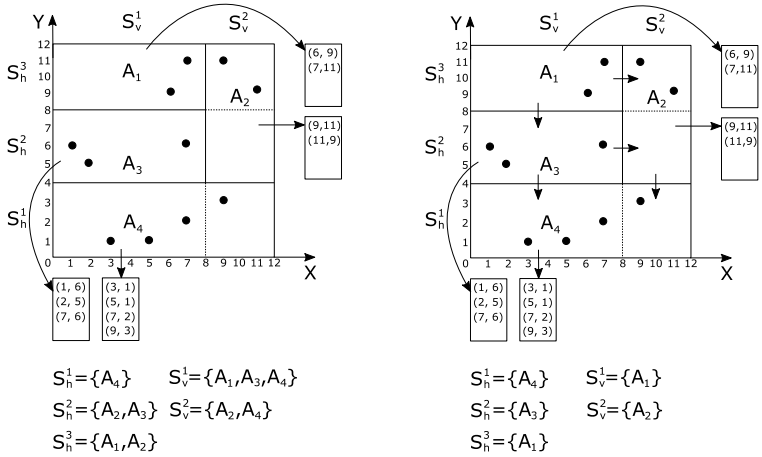


Рис. 3: Альтернативная структура сеточного файла

Доказано, что размер директории индекса, построенного согласно данному определению, имеет порядок $O(n^2 m)$. Произведено сравнение полученной оценки размера директории индекса с двумя методами организации сеточных файлов – многомерным динамическим хешированием (Multidimensional Dynamic Hashing – MDH) и многомерным расширяемым хешированием (Multidimensional Extensible Hashing – MEH)[18]. Показано, что в предлагаемом подходе к организации сеточных файлов размер директории индекса по сравнению с

[18] M. Regnier, "Analysis of Grid File Algorithms", *BIT*, vol. 25, no. 2, pp. 335-358, 1985

подходами MDH и MEH меньше в $\frac{sr^{\frac{1}{s}}}{n}$ и $\frac{sr^{\frac{n-1}{ns-1}}}{n}$ раз соответственно, где r есть количество записей, s – размер блока, а n – количество измерений.

Предлагаемая структура индекса позволяет естественным образом представить сеточный файл в виде ориентированного ациклического графа. Подобное представление используется в качестве реализации индекса.

Определение 3.2.2. Скажем, что граф $G = \langle V, E \rangle$ представляет сеточный файл $F = \langle D, S, C \rangle$, имеющий множество указателей R, если он построен согласно следующим условиям:

1. Для каждого сегмента $a \in C$ существует соответствующая ему вершина $v_a \in V$;
2. Для каждой полосы $s \in S$ существует соответствующая ей вершина $v_s \in V$.
3. Для каждого указателя $(a, b) \in R$ существует ребро $(v_a, v_b) \in E$, где a – полоса либо сегмент, b – сегмент.

На рис. 4 приведен пример сеточного файла и соответствующего ему графа.

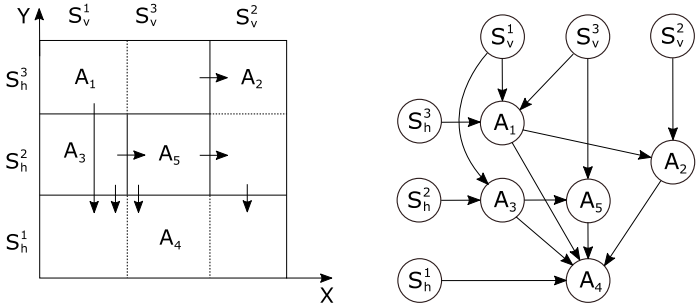


Рис. 4. Представление сеточного файла в виде графа

В §3.3 приведена формализация концепции сеточного файла с помощью XML. Целью XML-формализации концепции сеточного файла является создание языка определения директории, независимого от парадигм управления данными. Формализация концепции сеточного файла посредством XML предполагает разработку XML приложения. В этом приложении понятия измерения, полосы и сегмента сеточного файла определяются как XML элементы. XML-представление концепции сеточного файла определяется при помощи соответствующей DTD. На рис. 5 приведен пример представления в виде графа XML схемы, удовлетворяющей данной DTD. Метки ребер графа на этом рисунке имеют две функции, объединяющие информацию, содержащуюся в элементах и объявлениях отношений. Пусть ребро, ведущее из вершины *N* в вершину *M*, имеет метку *L*.

1. Вершина *N* может быть рассмотрена как объект или структура, а вершина *M* как один из атрибутов объекта или поле структуры. В этом случае *L* является именем атрибута или поля соответственно.
2. Вершины *N* и *M* могут быть рассмотрены как объекты, а *L* – как отношение между ними.

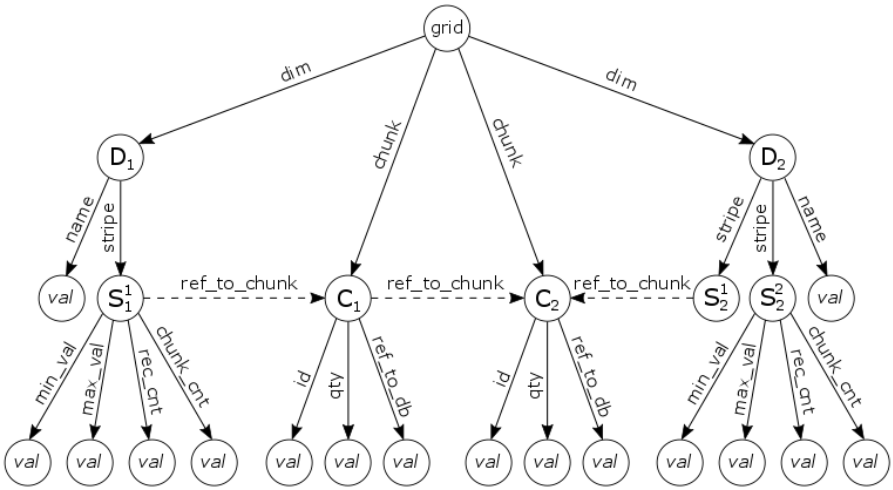


Рис. 5. Граф XML DTD

Четвертая глава диссертации посвящена разработке программной реализации и экспериментальным исследованиям. На основе предложенной в главе 3 динамической структуры индекса для многомерных данных был реализован прототип хранилища данных. Для реализации программного обеспечения был использован язык C++. Был проведен ряд экспериментов для сравнения производительности построенного прототипа с MongoDB[19]. MongoDB была выбрана для сравнения по прагматическим соображениям, т.к. является на сегодняшний день одной из наиболее востребованных NoSQL баз данных.

В §4.1 описана архитектура реализованного прототипа хранилища данных. Приведено детальное описание используемых классов с информацией об их функциональности, предоставляемом интерфейсе и используемых алгоритмах и структурах данных.

В §4.2 приведены результаты проведенных экспериментальных исследований. Для осуществления экспериментов были рассмотрены точки в трехмерном евклидовом пространстве, координаты точек представлены 32-битными беззнаковыми целыми числами. Эксперименты были осуществлены с использованием операций вставки, а также четырех основных категорий запросов[20] – поиск заданной точки, запросы к данным с совпадением отдельных координат, запросы в диапазонах значений, поиск ближайших соседних объектов. Далее приведены некоторые результаты проведенных экспериментов.

[19] https://www.mongodb.org/

[20] H. Garcia-Molina, J. Ullman and J. Widom, Database Systems: The Complete Book, Prentice Hall, 2009.

На рис. 6 (a) представлены графики роста размера директории индекса при осуществлении 2 млн. операций вставки. Ось абсцисс представляет количество хранимых в базе данных записей, а ось ординат – расходуемая индексом память (в Мб). На рис. 6 (b) изображены графики, отражающие время выполнения (в сек.) 10^5 операций поиска заданной точки в зависимости от количества хранимых записей. Можно видеть, что предлагаемая динамическая структура индекса занимает гораздо меньше памяти, чем используемые MongoDB B-деревья, а также имеет более быструю скорость обработки запросов поиска заданной точки.

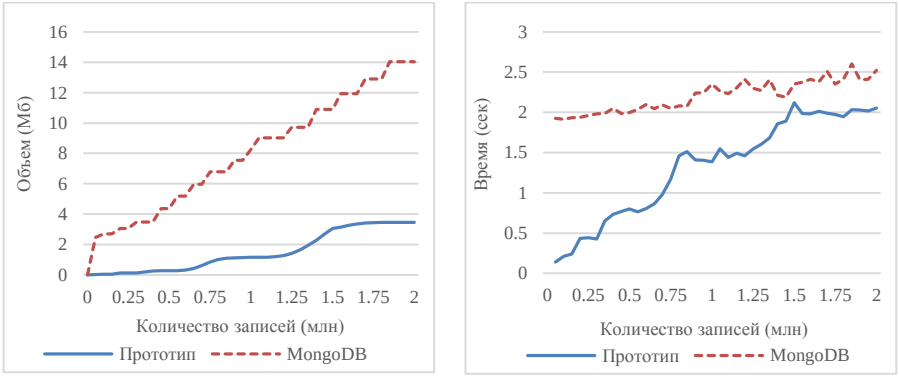


Рис. 6. (a) Размер директории индекса; (b) Поиск заданной точки

Запрос в диапазоне значений определяет прямоугольную область сетки, и ответом на него является множество точек, принадлежащих сегментам, которые покрывают эту область. На рис. 7 приведены графики, отражающие время обработки 10^5 запросов в случаях, когда заданные диапазоны имеют среднюю длину 10^3 и 10^6. Можно видеть, что при увеличении диапазонов поиска быстродействие построенного прототипа хранилища относительно MongoDB значительно возрастает.

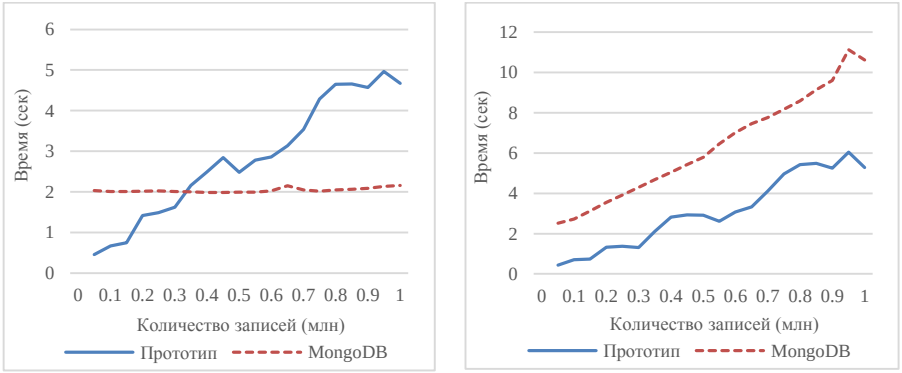


Рис. 7. (a) Поиск в диапазонах средней длины 10^3; (b) Поиск в диапазонах средней длины 10^6

На основе проведенных экспериментов показана эффективность построенного прототипа хранилища по сравнению с MongoDB в случаях поиска заданной точки, поиска в широких диапазонах значений, поиска ближайших соседних значений, а также более эффективное использование памяти.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ ДИССЕРТАЦИОННОЙ РАБОТЫ

1. Разработан подход к интеграции данных на основе принципа коммутативных отображений моделей данных;
2. Предложена расширяемая XML-каноническая модель данных;
3. Построены AMN-машины для канонической и реляционной моделей данных и их обратимого отображения и доказана его корректность;
4. Разработан подход к построению хранилища данных, основанный на динамической структуре индекса для многомерных данных, предложены эффективные алгоритмы для ее поддержки и приведены оценки сложности предложенных алгоритмов;
5. Разработан и реализован прототип хранилища данных на основе предложенной динамической структуры индекса.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ ПО ТЕМЕ ДИССЕРТАЦИИ

[1] М. Г. Манукян и Г. Р. Геворгян, “Об одном подходе к задаче интеграции информации”, *Сборник статей VI Годичной Научной Конференции РАУ*, с. 85-93, 2011.

[2] M. G. Manukyan and G. R. Gevorgyan, “An XML Mediator Based on the AMN Formalism”, *Russian-Armenian (Slavonic) University Bulletin*, vol. 1, pp. 3-18, 2012.

[3] M. G. Manukyan and G. R. Gevorgyan, “An Approach to Information Integration Based on the AMN Formalism”, in *Proceedings of First Workshop on Programming the Semantic Web*, Boston, 2012.

[4] G. R. Gevorgyan, “An Approach to Support Grid Files”, in *Proceedings of X Annual Scientific Conference of RAU*, 2015.

[5] G. R. Gevorgyan and M. G. Manukyan, “Effective Algorithms to Support Grid Files”, *Russian-Armenian (Slavonic) University Bulletin*, vol. 2, pp. 22-38, 2015.

[6] G. R. Gevorgyan, “An Effective Dynamic Structure for Grid File Organization”, *Russian-Armenian (Slavonic) University Bulletin*, vol. 1, pp. 5-17, 2016.

Գրիգոր Ռուբենի Գևորգյան

Տվյալների ինտեգրման մի մոտեցման մասին՝ մոդել, ալգորիթմներ և վերիֆիկացիա

Ամփոփում

Խնդրի արդիականությունը

Տվյալների ինտեգրման նպատակն է օգտագործել երկու կամ ավել տվյալների բազաներ (տվյալների աղբյուրներ) մի մեծ (հնարավոր է՝ վիրտուալ) տվյալների բազայի շրջանակում և դրան դիմել հարցումներով՝ որպես ունիֆիկացված ինֆորմացիոն տարածության: Ինֆորմացիայի ինտեգրման խնդիրները լուծելու համար օգտագործվող հիմնական տեխնոլոգիաներն են ֆեդերատիվ տվյալների բազաները, մեդիատորները և տվյալների պահոցները:

Ֆեդերատիվ տվյալների բազաների տեխնոլոգիան ենթադրում է, որ տվյալների աղբյուրները անկախ են, սակայն դրանցից յուրաքանչյուրը կարող է ստանալ անհրաժեշտ ինֆորմացիա մնացածներից: Ֆեդերատիվ տվյալների բազաների տեխնոլոգիաներ օգտագործող ժամանակակից կիրառություններ են IBM DB2-ը, Oracle Data Integrator-ը և GoldenGate-ը:

Տվյալների պահոցներ օգտագործելիս մի քանի տվյալների աղբյուրներից ինֆորմացիայի ֆրագմենտների կրկնօրինակներ պահվում են մի տվյալների բազայում: Տվյալների պահոցների հենքի վրա կառուցված տվյալների ինտեգրման կարևոր կիրառությունների դասեր են հանդիսանում տվյալների առցանց մշակման և դրանց ինտելեկտուալ վերլուծության կիրառությունները:

Մեդիատորները իրենցից ներկայացնում են ծրագրային բաղադրիչներ, որոնք ապահովում են վիրտուալ տվյալների բազաների կոնցեպցիան: Վիրտուալ տվյալների ինտեգրման բնագավառում ժամանակակից հետազոտություններից կարելի է առանձնացնել մասնավորապես SYNTHESIS խմբի աշխատանքները (որոնք պիոներներն են անհամասեռ տվյալների բազաների համար տվյալների մոդելների հիմնավորված արտապատկերումների բնագավառում) և Պաոլո Ատցենիի հետազոտությունները, որոնք կողմնորոշված են ինչպես կառուցվածքային, այնպես էլ NoSQL տվյալների բազաների ինտեգրման խնդիրներին:

Գիտության և ինֆորմացիոն տեխնոլոգիաների բնագավառում նոր պարադիգմի առաջացումը կապված է գերմեծ տվյալների ղեկավարման խնդիրների հետ: Այս բնագավառում հիմնական խնդիրներից է անհամասեռ տվյալների աղբյուրների ինտեգրացիան:

Աշխատանքի նպատակն է հետազոտել տվյալների ինտեգրման խնդիրները, կառուցել տվյալների ինտեգրման մոտեցում՝ հենված տվյալների մոդելների կոմուտատիվ

արտապատկերումների սկզբունքի վրա, ինչպես նաև մշակել կանոնական տվյալների մոդել, կատարել տվյալների մոդելների արտապատկերման կոռեկտության ստուգում, և առաջարկել վիրտուալ ու նյութականացված ինտեգրման աջակցման ալգորիթմներ:

Կիրառական նշանակությունը: Առաջարկված տվյալների ինտեգրման մոտեցումը կարող է օգտագործվել ինչպես տվյալների առցանց մշակման և դրանց ինտելեկտուալ վերլուծության կիրառություններում, այնպես էլ գերմեծ տվյալների բնագավառում:

Ատենախոսությունում ստացված են հետևյալ արդյունքները.

1. Մշակվել է տվյալների մոդելների կոմուտատիվ արտապատկերումների սկզբունքի հենքի վրա տվյալների ինտեգրման մոտեցում:
2. Առաջարկվել է ընդլայնվող XML-կանոնական տվյալների մոդել:
3. Կառուցվել են AMN-մեքենաներ կանոնական և ռելացիոն տվյալների մոդելերի և դրանց միջև հակադարձելի արտապատկերման համար, ապացուցվել է այդ արտապատկերման կոռեկտությունը:
4. Մշակվել է բազմաչափ տվյալների համար դինամիկ ինդեքսի կառուցվածքի հենքի վրա տվյալների պահոցի կառուցման մոտեցում, առաջարկվել են դրա աջակցման համար էֆեկտիվ ալգորիթմներ և բերվել են այդ ալգորիթմների գնահատականներն ու բարդությունները:
5. Առաջարկված դինամիկ ինդեքսի կառուցվածքի հենքի վրա մշակվել է տվյալների պահոցի նախատիպ:

Grigor Gevorgyan

An approach to data integration: model, algorithms and verification

Summary

Actuality of the problem

The aim of information integration is to use two or more databases (data sources) in the frame of a large database, possibly virtual, containing information from all sources, so the data can be queried as a unit. The main technologies, used to solve information integration problems, are *federated databases*, *mediators* and *data warehouses*.

Federated databases technology assumes that data sources are independent, but each of them can retrieve necessary information from the others. Examples of modern applications using federated databases technologies are IBM DB2, Oracle Data Integrator and GoldenGate.

Mediators represent software components that provide support of virtual databases. Mediator does not store information itself; instead it transmits the user request to one or more request, addressed

to primary sources. Among the current research in the area of virtual data integration we should distinguish the works SYNTHESIS group (IPI RAS) who are pioneers in the area of justifiable data models mapping for heterogeneous databases integration, research of Paolo Atzeni, which is oriented to integration of structured, as well as NoSQL databases.

When using data warehouses copies of information fragments from several sources are saved in a single database, possibly with preliminary processing. Important classes of information integration systems application based on data warehouses technologies are the *OLAP* (on-line analytic processing) and *data mining* applications.

The emergence of a new paradigm in science and various applications of information technology (IT) is related to issues of big data handling. One of the key problems in this area is integration of heterogeneous data sources.

The purpose of the work is to investigate problems of data integration, to develop an approach to data integration based on the data models commutative mappings principle, including development of canonical data model, method of data models mapping correctness verification, and algorithms for supporting virtual and materialized integration.

Practical significance. The proposed data integration method can be used for heterogeneous information sources integration in OLAP applications, data mining applications, and in the area of big data. Experiments conducted using the constructed data warehouse prototype show that usage of the proposed method can lead to significant reduction of index directory size and lookup time for multidimensional data.

The main results of the work are the following:

1. A method of data integration based on the principle of commutative mappings of data models is developed;
2. An extensible XML-canonical data model is proposed;
3. AMN-machines for canonical and relational data models and their reversible mapping are constructed, correctness of that mapping is proved;
4. A method of data warehouse construction, based on a dynamic index structure for multidimensional data, is developed, effective algorithms for its support are proposed and estimations of complexities for proposed algorithms are provided;
5. A data warehouse prototype based on the proposed dynamic index structure is designed and developed.