

Научная статья

Статья в открытом доступе

УДК 004.658.2

doi:10.30987/2658-6436-2022-3-78-83

ПОДХОД К ПОВЫШЕНИЮ ЭФФЕКТИВНОСТИ ПОСТРОЕНИЯ ПЕРСОНАЛИЗИРОВАННЫХ РЕКОМЕНДАЦИЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ В ОБЛАЧНЫХ СИСТЕМАХ ЭЛЕКТРОННОГО ОБУЧЕНИЯ

Татьяна Михайловна Геращенко¹, Дмитрий Иванович Гончаров²,
Андрей Олегович Маркелов³

^{1,2,3}Брянский государственный технический университет, г. Брянск, Россия

¹gerash-tatyana@yandex.ru

²jeriho32@yandex.ru

³andrmarkelov@yandex.ru

Аннотация. Целью данного исследования является разработка подхода к построению рекомендаций в системах электронного обучения с использованием алгоритмов графовой базы данных. В процессе исследования была поставлена задача: проанализировать подход использования в системах электронного обучения классических алгоритмов реляционных баз данных, выявить его недостатки и предложить новый подход с использованием алгоритмов графовых баз данных. В качестве методов исследования был использован анализ, синтез, сравнение и системный подход. Предложен новый подход к построению рекомендаций в режиме реального времени в системах электронного обучения с использованием алгоритмов графовых баз данных. В результате исследования были показаны ключевые преимущества разработанного подхода. Показано, как легко смоделировать домен рекомендаций в виде графа, взяв Cypher в качестве языка для получения информации из базы данных графов. Графовые базы данных позволяют мгновенно реагировать на действия пользователей и вычислять их рекомендации в режиме реального времени, что делает их рекомендации более персонализированными. В качестве примера применения был взят сценарий из системы электронного обучения.

Ключевые слова: социальная система, таджикские студенты, обучающиеся в вузах РФ, система управления

Для цитирования: Геращенко Т.М., Гончаров Д.И., Маркелов А.О. Подход к повышению эффективности построения персонализированных рекомендаций в режиме реального времени в облачных системах электронного обучения. 2022. №3 (17) С. 78-83 doi: 10.30987/2658-6436-2022-3-78-83.

Original article

Open Access Article

APPROACH TO IMPROVING THE EFFICIENCY OF BUILDING REAL- TIME PERSONALIZED RECOMMENDATIONS IN CLOUD ELEARNING SYSTEMS

Tatyana Mikhailovna Gerashchenkova¹, Dmitry Ivanovich Goncharov²,
Andrey Olegovich Markelov³

^{1,2,3}Bryansk State Technical University, Bryansk, Russia

¹gerash-tatyana@yandex.ru

²jeriho32@yandex.ru

³andrmarkelov@yandex.ru

Abstract. The aim of this study is to develop an approach to building recommendations in e-learning systems using graph database algorithms. In the course of the study, the task is to analyze the approach of using classical relational database algorithms in e-learning systems, identify its shortcomings and propose a new approach using graph database algorithms. Analysis, synthesis, comparison and a systematic approach are applied as the research methods. The article proposes a new approach to building realtime recommendations in e-learning systems using graph database algorithms. As the study result, the key advantages of the developed approach are presented. This article shows how easy it is to model a recommendation domain as a graph, using Cypher as the language for getting data from a graph database. Graph databases allow one to instantly respond to user actions and calculate their recommendations in real time, which makes their suggestions more personalized. A scenario from an e-learning system is taken as an application example.

Keywords: recommender systems, e-learning systems, graph databases

For citation: Gerashchenkova T.M., Goncharov D.I., Markelov A.O. Approach to improving the efficiency of building real-time personalized recommendations in cloud elearning systems. Automation and modeling in design and management, 2022, no. 3 (17). pp. 78-83. doi: 10.30987/2658-6436-2022-3-78-83

Введение

В статье представлен подход к проблеме построения рекомендаций, который заключается в использовании алгоритмов графовой базы данных. Кроме того, используя графовую структуру данных для хранения данных, мы можем легко перемещаться по графу, находя общие для пользователей закономерности.

Традиционные механизмы рекомендаций в системах электронного обучения работают в автономном режиме: процесс пакетной обработки (упаковка нескольких отдельных транзакций в один пакет) информации в базе данных пропускает историю пройденных учебных материалов пользователем через набор алгоритмов и генерирует персональные рекомендации с ранее настроенной в системе периодичностью. Для того чтобы понять недостатки пакетной обработки информации, предположим, что пользователь в системе электронного обучения добавляет к курсу по основам языка программирования Java еще один похожий курс по основам Java. Что в таком случае может быть предложено пользователю изучить в качестве следующего материала? В большинстве случаев – третий курс по основам Java [1, с. 351].

Такой механизм рекомендаций обладает информацией о прошлых курсах, которые изучил пользователь, но он не обладает механизмами, которые позволят составить рекомендацию по изучению следующих учебных материалов, которые могут быть интересны пользователю в качестве следующего шага по изучению языка программирования Java. В результате последующие рекомендации не будут интересны пользователю, а он, в свою очередь, с большой долей вероятности, проигнорирует их.

Лучшим подходом было бы реагировать на действия пользователя в системе, пересчитывать рекомендации в режиме реального времени, пока он добавляет различные курсы в свой профиль и предлагать учебные материалы в соответствии с тем, что уже изучил пользователь и добавил в свой профиль для изучения в будущем. Это даст ощущение пользователю, что он взаимодействует с ментором и сделает этот процесс более персонализированным.

Материалы, модели, эксперименты и методы

Исследование построения рекомендаций в режиме реального времени был разбит на несколько этапов. На первом этапе был произведен анализ классического подхода к построению рекомендаций на примере облачных систем электронного обучения. Далее были выявлены и продемонстрированы недостатки данного подхода. На следующем этапе был предложен новый подход к построению рекомендаций в режиме реального времени с использованием алгоритмов графовых баз данных. Были показаны преимущества предложенного подхода на примере облачных систем электронного обучения на больших массивах данных и его сравнение с классическим подходом. Далее схематично были продемонстрированы примеры работы данного алгоритма с использованием языка запросов Cypher.

Результаты

Основной принцип всех алгоритмов рекомендаций одинаков: они начинают с поиска объектов с похожими свойствами и вычисления меры их «сходства». Например, пользователи, которые живут в одном районе, имеют одинаковый возраст или общие интересы, «группируются вместе».

На втором этапе необходимо проанализировать все действия, совершенные каждым пользователем. Зная предыдущие действия двух похожих пользователей, можно найти для них подходящие рекомендации.

Представим, что Алиса прошла курс-1 и курс-2. Если Боб также прошел курс-1, и модель рекомендаций оценивает его как похожего на Алису, система порекомендует ему пройти курс-2.

Ключевым моментом здесь является то, что оба этапа процесса зависят от отношений: во-первых, от отношений между пользователями, а во-вторых, от отношений между пользователями и их курсами. Чем быстрее будет возможность запрашивать и просматривать эти

отношения, тем больше возможностей для предоставления рекомендаций в режиме реального времени.

Реляционные базы данных могут моделировать отношения, но для того, чтобы проследить эти отношения, необходимо написать SQL-запросы, объединяющие таблицы. Процесс соединения требует больших вычислительных затрат и становится медленнее по мере увеличения количества соединений, что делает анализ в режиме реального времени непрактичным в продуктивных средах [1, с. 527].

Графовые базы данных могут помочь решить эту проблему. Графовая база данных – это система управления, работающая на графовой модели данных. В отличие от других баз данных, в графовых базах данных приоритет отдается отношениям. Это означает, что системе не нужно выводить связи между данными, используя такие вещи, как внешние ключи.

Графовые данные – это представление, использование и сохранение связей между элементами данных. Ключевым моментом здесь является сохранение информации об отношениях, а не просто их установка, как это может быть сделано с помощью таблицы JOIN в реляционной системе управления базами данных (рис. 1).

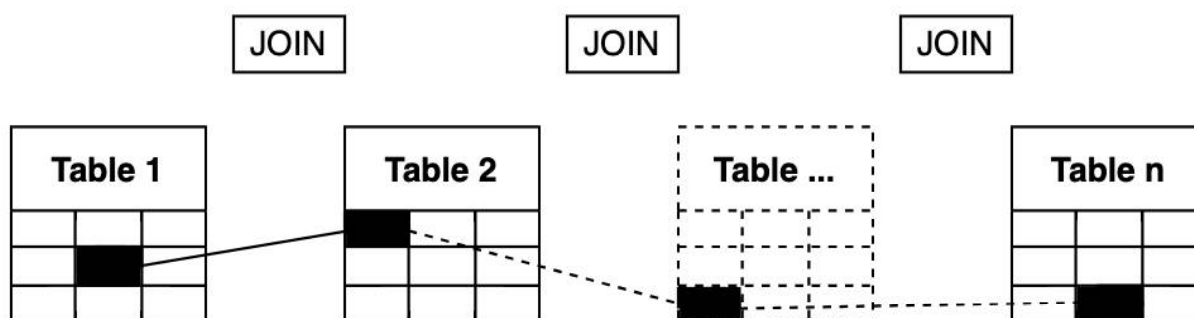


Рис. 1. Схема выявления отношений в классических реляционных базах данных
Fig. 1. Scheme of identifying relationships in classical relational databases

Поскольку отношения явно выражены элементами ребер, перемещение по графу от одной вершины к другой концептуально просто и не требует больших вычислительных затрат.

В результате можно выполнять запросы на основе отношений в режиме реального времени, быстро запрашивать данные о прошлых курсах, пройденных пользователями, а также мгновенно фиксировать любые новые интересы, проявившиеся во время их текущего взаимодействия с электронной системой обучения, что очень важно для составления рекомендаций в режиме реального времени.

Контекст – это любая информация, которая может быть использована для характеристики условий, в которой находится субъект. Субъект – это человек, место или объект, который считается релевантным для взаимодействия между пользователями и предметом. Наиболее распространенными типами контекста являются личность («кто»), деятельность («что»), время («когда») и местоположение («где»), и с помощью этой информации можно определить, «почему» возникает ситуация.

Графовые базы данных могут сделать рекомендации более персонализированными за счет включения контекстной информации, используя связи между данными. Этого невозможно достичь с помощью технологии реляционных баз данных, поскольку SQL-запросы слишком сложны и требуют слишком много времени для выполнения [1, с. 32].

На примере разрабатываемого модуля рекомендации для системы электронного обучения раскрыто представление о том, как легко генерировать основанные на графе персонализированные рекомендации курсов в режиме реального времени в системе электронного обучения.

Будет использован язык запросов Cypher, который является декларативным языком запросов к графу, позволяющим выполнять эффективные запросы и обновления графа свойств [3, с. 52].

В сценарии электронного обучения мы можем хранить профили, в которых сохранена информация о пройденных и только добавленных курсах, в графе, подобном приведенному ниже.

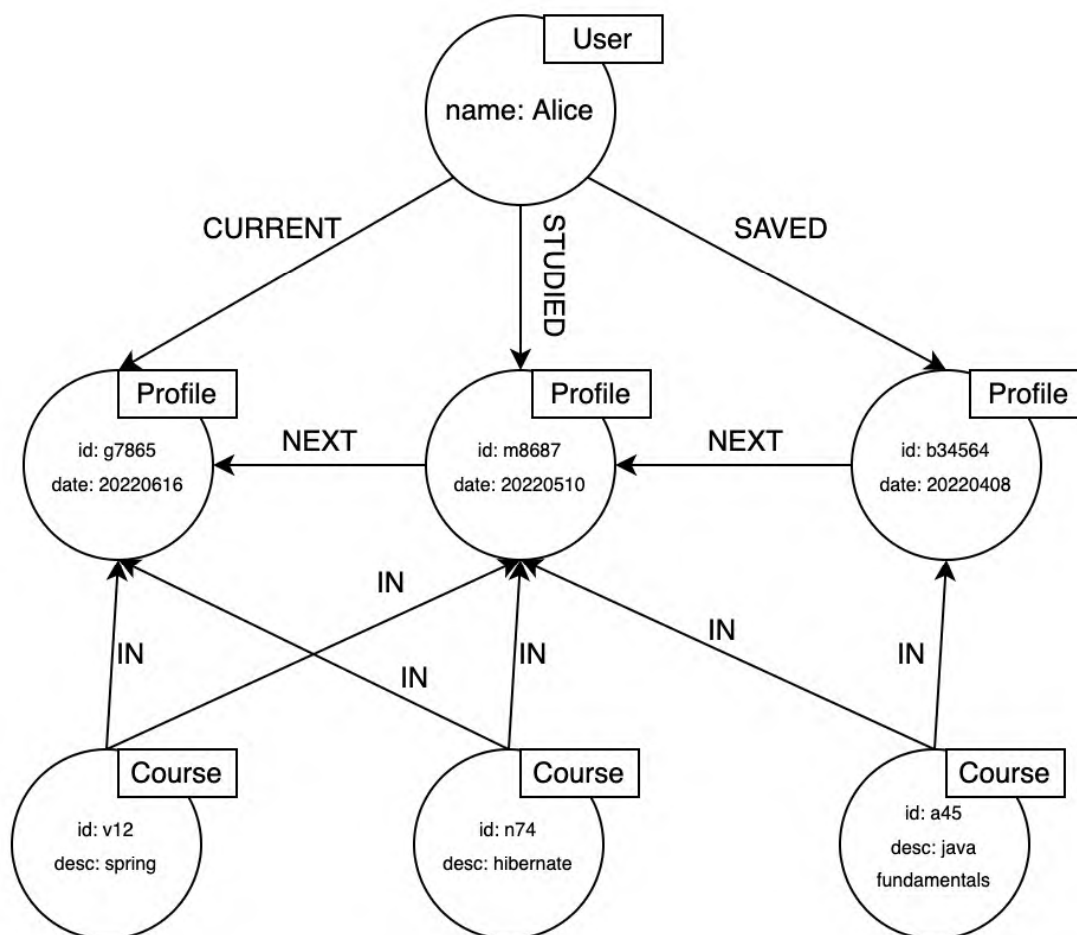


Рис. 2. Граф, отображающий историю взаимодействия пользователя с системой электронного обучения
 Fig. 2. Graph showing the history of user interaction with the e-learning system

Приведенный выше граф показывает, как можно создать историю пройденных курсов для пользователя, построив простой связанный список профилей, соединенных отношениями NEXT.

На этом графе видно, что пользователь посетил систему обучения три раза, сохранил свой первый профиль, чтобы в дальнейшем пройти курсы из него (связь SAVED между узлами клиент и профиль), в конечном итоге прошел курсы из второго профиля (на это указывает связь STUDIED между узлом клиент и корзина) и в настоящее время собирает текущий профиль, что показано связью CURRENT.

Важно понимать, что это не схема или ER-диаграмма, а фактические данные одного пользователя.

В виде графа легко понять поведение пользователя: он стал новым пользователем, но не смог решиться на прохождение курса по Java fundamentals вернулся через месяц, изучив уроки по Java fundamentals, Hibernate и Spring. В итоге пользователь остановился на изучении Hibernate и Spring.

Теперь, когда имеется граф, есть возможность подумать о рекомендациях, чтобы повлиять на будущее взаимодействие пользователя с системой электронного обучения. Самая простая рекомендация – это показать популярные курсы в системе электронного обучения. Это тривиально в CypHer, как будет показано в следующем запросе.

MATCH

(user:User)-[:STUDIED]->(:Profile) <-[:IN]-(course:Course)

```
RETURN course, count(course)
ORDER BY count(course) DESC LIMIT 5
```

Синтаксис Cypher обеспечивает визуальный и логический способ сопоставления паттернов узлов и связей в графе. Это декларативный, последовавший за SQL язык для описания визуальных паттернов в графах с использованием синтаксиса ASCII-Art. Он позволяет указать, что необходимо выбрать из данных графа, без описания того, как именно это сделать.

Однако в приведенном выше запросе нет реального контекста. Постараемся достичь большего без избыточных операций, рекомендуя популярные образовательные материалы, на базе ранее изученных пользователем, как в следующем запросе:

```
MATCH
(user:User {name: 'Alice'})-[:STUDIED]->(:Profile) <-[:IN]-(course:Course)
RETURN course, count(course)
ORDER BY count(course) DESC LIMIT 5
```

Единственным изменением в этом запросе по сравнению с предыдущим является включение ограничения на узел «user».

Как объяснялось ранее, можно еще детальнее персонализировать рекомендации, включив в них контекстную информацию. Помимо этого, можно акцентировать внимание пользователя на популярных курсах в его «социальном круге». Как и следовало ожидать, добавить социальную информацию в граф легко, а запрос друзей/знакомых/соседей/одногруппников или других данных о социальном круге прост.

```
MATCH
(user:User {name: 'Alice'})-[:FRIEND*1..2]->(friend:User) WHERE user <> friend
WITH DISTINCT friend
MATCH
(friend)-[:STUDIED]->(:Profile) <-[:IN]-(course:Course)
RETURN course, count(course)
ORDER BY count(course) DESC LIMIT 5
```

Таким образом, мы можем рекомендовать наиболее популярные курсы друзей или однокурсников/одногруппников, опираясь на имеющиеся место связи между данными.

Заключение

Традиционные рекомендательные системы изначально создавались без устоявшейся технологии быстрого и эффективного анализа взаимосвязей между данными. Запрос и обход этих связей требует больших вычислительных затрат, что делает анализ в режиме реального времени неэффективным в больших масштабах.

Было показано, как легко смоделировать домен рекомендаций в виде графа, взяв Cypher в качестве языка для получения данных из базы данных графов.

Графовые базы данных позволяют мгновенно реагировать на действия пользователей и вычислять их рекомендации в режиме реального времени, что делает эти рекомендации более персонализированными. В качестве примера применения был взят сценарий из системы электронного обучения.

Список источников:

References:

1. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика - М.: Вильямс И.Д., 2017. - 1440 с.
2. Anant Kumar. Graph Database Modeling with neo4j, Independently published; 2020, 124 p.
3. Cristian Scutaru. Advanced Neo4j Cypher Puzzles with Answers, Independently published; 2020, 110 p.
1. Connolly T. Databases. Design, Implementation and Maintenance. Theory and Practice. Moscow: Williams Publishing House; 2017.
2. Kumar An. Graph Database Modelling with neo4j. Wydawca: Independently published; 2020.
3. Scutaru Cr. Advanced Neo4j Cypher Puzzles with Answers. Independently published; 2020.

4. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: Методы и средства проектирования информационных систем и технологий / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. - М.: Форум, 2017. - 62 с.

5. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: Методы и средства проектирования информационных систем и технологий / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. - М.: Форум, 2018. - 61 с.

6. Базы данных: проектирование: Учебник для академического бакалавриата / Н.П. Стружкин, В.В. Годин. - Люберцы: Юрайт, 2016. -477 с.

4. Martishin S.A., Simonov V.L., Khrapchenko M.V. Designing and Implementing Databases in MySQL DBMS Using MySQL Workbench: Methods and Tools for Designing Information Systems and Technologies. Moscow: Forum; 2017.

5. Martishin S.A., Simonov V.L., Khrapchenko M.V. Designing and Implementing Databases in MySQL DBMS Using MySQL Workbench: Methods and Tools for Designing Information Systems and Technologies. Moscow: Forum; 2018.

6. Struzhkin N.P., Godin V.V. Databases: Designing. Lyubertsy: Yurayt; 2016.

Информация об авторах

Татьяна Михайловна Геращенко

доктор экономических наук, доцент «Брянский государственный технический университет», г. Брянск, Россия

Дмитрий Иванович Гончаров

аспирант «Брянский государственный технический университет», г. Брянск, Россия

Андрей Олегович Маркелов

аспирант «Брянский государственный технический университет»

Information about authors:

Gerashchenkova Tatyana Mikhailovna

Doctor of Economics, Associate Professor of Bryansk State Technical University

Goncharov Dmitry Ivanovich

post-graduate student of Bryansk State Technical University

Markelov Andrey Olegovich

post-graduate student of Bryansk State Technical University

Вклад авторов: Геращенко Т.М. – 0,2; Гончаров Д.И. – 0,6; Маркелов А.О. – 0,2.

Contribution of the authors: Gerashchenkova T.M. – 0.2; Goncharov D.I. - 0.6; Markelov A.O. - 0.2.

Авторы заявляют об отсутствии конфликта интересов.

The authors declare no conflicts of interests.

Статья поступила в редакцию 29.06.2022; одобрена после рецензирования 02.09.2022; принята к публикации 09.09.2022.

The article was submitted 29.06.2022; approved after reviewing 02.09.2022; accepted for publication 09.09.2022.

Рецензент – Фраймович Д.Ю., доктор экономических наук, профессор, Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых.

Reviewer – Fraimovich D.Yu., Doctor of Economic Sciences, Professor, Vladimir State University named after Alexander and Nikolay Stoletovs.