

Заключение. При помощи реализованной программы на языке программирования C++ была исследована эффективность использования распараллеливания двумерной квазилинейной задачи теплопроводности. Исследования показали, что использование распараллеливания для решения задачи теплопроводности методом переменных направлений является эффективным и может быть использовано для сокращения времени вычислений. Практическая значимость исследований состоит в том, что результаты могут быть использованы в учебном процессе для студентов ИТ-специальностей.

Литература:

1. Кормен Т. Алгоритмы: построение и анализ. / Пер. с англ. / Т.Кормен и др. – М.: ООО «И.Д.Вильямс», 2013. – 1328 с.
2. Макконнелл, С. Совершенный код. Мастер-класс / Пер. с англ. / С. Макконнелл. – СПб.: Издательско-торговый дом «Русская редакция», 2005. – 896 с.
3. Анотонов, А.С. Параллельное программирование с использованием технологии MPI: Пособие / Анотонов А.С. – М.: Изд-во МГУ, 2004. – 71 с.
4. Воеводин, В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин – СПб.: БХВ-Петербург, 2002. – 608 с.

АРХИТЕКТУРА СИСТЕМЫ АНАЛИЗА И КОНСОЛИДАЦИИ ДАННЫХ ИЕРАРХИЧЕСКИХ ОБЪЕКТОВ

Соколовский В.В., Шестаков И.В.,

студенты 5 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь

Научный руководитель – Ермоченко С.А., канд. физ.-мат. наук

На сегодняшний день, ценность и достоверность знаний, полученных в результате интеллектуального анализа бизнес-данных, зависит не только от эффективности используемых аналитических методов и алгоритмов, но и от того, насколько правильно подобраны и подготовлены исходные данные для анализа.

Обычно руководителям проектов по бизнес-аналитике приходится сталкиваться со следующими проблемами. Во-первых, данные на предприятии расположены в различных источниках самых разнообразных форматов и типов – в отдельных файлах офисных документов (Excel, Word, обычных текстовых файлах), в учетных системах («1С: Предприятие», «Парус» и др.), в базах данных (Oracle, Access, dBase и др.). Во-вторых, данные могут быть избыточными или, наоборот, недостаточными. А в-третьих, данные являются «грязными», то есть содержат факторы, мешающие их правильной обработке и анализу (пропуски, аномальные значения, дубликаты и противоречия).

Поэтому, прежде чем приступать к анализу данных, необходимо выполнить ряд процедур, цель которых – доведение данных до приемлемого уровня качества и информативности, а также организовать их интегрированное хранение в структурах, обеспечивающих их целостность, непротиворечивость, высокую скорость и гибкость выполнения аналитических запросов.

Зачастую такая задача решается с применением технологии OLAP и соответствующих коммерческих систем [1]. Однако, помимо высокой стоимости таких решений, они часто оказываются довольно громоздкими и ориентированы на обработку больших объемов данных, что не всегда оправдано в небольших системах.

Целью работы является проектирование архитектуры для системы анализа и консолидации данных иерархических объектов с небольшим объемом информации.

Материал и методы. Материалом исследования является набор иерархических объектов и их числовых параметров.

Для создания web-приложения использовались следующие технологии:

- Серверная часть:

- Язык программирования Java, JavaScript.
- Контейнер сервлетов Tomcat, сервер NodeJS.
- Фреймворк Spring, библиотека Hibernate.
- СУБД MySQL.
- БД MariaDB.

- Клиентская часть:

- Языки программирования JavaScript, TypeScript.
- Язык разметки HTML.
- Язык написания каскадных таблиц стилей CSS3.
- Фреймворк AngularJS, библиотека D3.js, библиотека Bootstrap.

Методы исследования – моделирование, анализ, методы объектно-ориентированного и прототипно-ориентированного программирования, ранжирование, тестирование.

Результаты и их обсуждение. В основу архитектуры приложения была положена иерархия объектов. Каждый объект в иерархии имеет некий набор параметров. Так как целью работы является разработка системы анализа и консолидации данных без привязки к конкретной предметной области, то и

природа иерархии объектов нам не важна, как и набор параметров этих объектов. Система позволяет создавать любые объекты с произвольными числовыми параметрами.

Для анализа значений параметров имеющихся объектов задаются формулы, описывающие правила консолидации данных верхних уровней иерархии на базе значений параметров с нижних уровней. Параметры объектов могут изменяться с течением времени. Для удобства анализа изменения параметров во времени на разных уровнях иерархии система должна позволять автоматически формировать различные графические диаграммы [2].

На основании всего вышеизложенного была спроектирована структура базы данных для хранения иерархии объектов и значений их параметров. Обработка и визуализация данных осуществляется с применением современных web-технологий. На основе спроектированной архитектуры было создано web-приложение, позволяющее пользователю создавать и визуализировать иерархическую структуру объектов произвольных данных, исходя из выбранных пользователем настроек.

В результате анализа было выявлено, что реляционные базы данных с увеличением объема данных и с увеличением связности между ними, имеют большую избыточность при хранении, а также теряют в скорости обработки запросов. Также для хранения формул для вычисления некоторых параметров объектов использовалась обратная польская запись. Хранение формул в реляционной БД вызывало еще большую избыточность.

В связи с проблемами описанными выше было принято решение использовать NoSQL базы данных (MongoDB) [3].

В качестве оптимизации при работе с сетью было решено использовать NodeJS сервера [4], ориентированные на прием запросов от клиента с дальнейшей пересылкой на другие внутренние сервера, выполняющие операции бизнес логики, передачу статических файлов, хранение кэша. Еще одним плюсом использования NodeJS стала его высокая интегрированность с клиентской частью, в основном за счет использования одного языка программирования.

Заключение. В результате работы была построена архитектура для системы анализа и консолидации данных иерархических объектов, а также создано приложение, которое служит для анализа иерархий произвольных объектов. В рамках созданной системы анализа данных:

- спроектирована архитектура серверной части, отвечающей за обработку данных.
- спроектирована БД для хранения данных.
- спроектирована архитектура клиентской части, отвечающей за визуализацию данных.
- проведено тестирование системы.
- произведен анализ полученных результатов.

Литература:

1. Codd, Edgar F. Providing OLAP to User-Analysts: An IT Mandate // Computerworld. — Т. 27, № 30.
2. D3. D3 API Reference [Электронный ресурс удалённого доступа] / Режим доступа: <https://github.com/d3/d3/blob/master/API.md>; Дата доступа: 27.02.2017.
3. David Hows, Peter Membrey, Eelco Plugge, Tim Hawkins. The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB, Third Edition. — Apress, 2015. — 376 с.
4. Node.js Foundation. Node.js v7.6.0 Documentation [Электронный ресурс удалённого доступа] / Режим доступа: <https://nodejs.org/api/>; Дата доступа: 28.02.2017.

О СТОУНОВЫХ РЕШЕТКАХ n -КРАТНО ω -ЛОКАЛЬНЫХ КЛАССОВ ФИТТИНГА

Титова А.И.,

магистрант ВГУ имени П.М. Машиерова, г. Витебск, Республика Беларусь

Научный руководитель – Воробьев Н.Н., доктор физ.-мат. наук, доцент

Все рассматриваемые группы конечны. Мы будем использовать стандартную терминологию из [1, 2].

Напомним, что класс групп $\mathfrak{F}$ называется классом Фиттинга, если он замкнут относительно нормальных подгрупп и произведений нормальных $\mathfrak{F}$ -подгрупп.

Пусть ω – некоторое непустое множество простых чисел $\omega' = \mathbb{P} \setminus \omega$. Через $\pi(G)$ обозначено множество всех простых делителей порядка группы G . Символы (1) , $\mathfrak{N}$, $\mathfrak{N}_p$, $\mathfrak{G}_p$ и $\mathfrak{G}_{\omega d}$ обозначают соответственно класс всех единичных групп, класс всех нильпотентных групп, класс всех p -групп, класс всех p' -групп и класс всех таких групп, у которых каждый композиционный фактор является ωd -группой.

Напомним, что для произвольного класса групп $\mathfrak{F} \supseteq (1)$ символ $G^{\mathfrak{F}}$ обозначает пересечение всех таких нормальных подгрупп N из G , что $G/N \in \mathfrak{F}$. Полагают (см. [3]), что $G^{\omega d} = G^{\mathfrak{G}_{\omega d}}$, $F^p(G) = G^{\mathfrak{N}_p \mathfrak{G}_{p'}}$.

Пусть f – произвольная функция вида

$$f : \omega \cup \{\omega'\} \rightarrow \{\text{классы Фиттинга}\}, \quad (*)$$

где $f(\omega') \neq \emptyset$. Функции f сопоставляют класс групп

$$LR_{\omega}(f) = (G \mid G^{\omega d} \in f(\omega') \text{ и } F^p(G) \in f(p) \text{ для всех } p \in \omega \cap \pi(G)).$$