

ПРИНЦИПЫ ПРЕДСТАВЛЕНИЯ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ В ГРАФОВЫХ БАЗАХ ДАННЫХ

Лось П. В.

Кафедра интеллектуальных информационных технологий,
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: p.los@bsuir.by

Анализируется и описывается возможный вариант представления семантических моделей в графовых базах данных с использованием таких технологий, как Neo4j и Apache Kafka.

ВВЕДЕНИЕ

Семантические модели в наши дни являются достаточно распространенной формой хранения и обработки информации в базах знаний интеллектуальных информационных систем. Семантическая паутина [1] и технология OSTIS [2] – лишь немногие примеры, где используется данный подход. Рассмотрим семантические модели немного подробнее.

I. АНАЛИЗ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ И НАПРАВЛЕНИЙ ИХ ПРИМЕНЕНИЯ

Семантические модели, как понятно из названия, для представления информации (знаний) используют семантические сети – это ориентированный граф, вершины которого – понятия, а дуги – отношения между ними. В качестве понятий обычно выступают абстрактные или конкретные объекты, а отношения – это связи типа: «это», «имеет частью», «принадлежит» и др. Характерной особенностью семантических сетей является обязательное наличие трех типов отношений:

- класс – элемент класса;
- свойство – значение;
- пример элемента класса.

Семантическая паутина для представления таких графов использует URI – унифицированный идентификатор ресурса или адрес, используемый для указания ссылок на какой-либо объект. В основном вся информация представляется в виде текста, который больше подходит для восприятия людьми, но не для машинной обработки.

Для того, чтобы сделать их удобными для «машин», были предложены RDF, RDF Schema, OWL – язык описания семантической сети и два языка описания онтологий соответственно.

Иным примером технологии на основе семантических моделей является OSTIS. Она представляет собой технологию компонентного проектирования семантически совместимых интеллектуальных систем, содержащих базы знаний любого уровня сложности и реализующих параллельные модели обработки знаний. База знаний ostis-системы представляет собой семантическую сеть, имеющую сложную иерархическую струк-

туру, которая состоит из элементов, обозначающих не только внешние сущности и связи между ними, но и различные классы элементов семантической сети, связи между классами и фрагментами.

Языком представления знаний является SC-код, который позволяет описывать структуру любой информационной конструкции и хорошо приспособлен к использованию в условиях нестационарности и неточности, противоречивости, неактуальности, неполноты знаний. Существует несколько форм внешнего представления для SC-кода: SCn (natural), SCg (graphical), SCs (string) – они служат для представления знаний в удобной форме в зависимости от контекста.

II. ПРОБЛЕМЫ ХРАНЕНИЯ И ОБРАБОТКИ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ НА СОВРЕМЕННЫХ КОМПЬЮТЕРАХ

Современные компьютеры не всегда хорошо подходят для интерпретации семантических моделей [3]. Это связано с низкой производительностью традиционных систем по причине последовательной обработки информации, низкого уровня доступа к памяти, её линейной организации и еще по ряду причин.

SC-код, который используется технологией OSTIS в качестве языка представления знаний, ориентирован на компьютеры нового поколения, в которых решены выше упомянутые проблемы, но таковые на данный момент не существуют. В связи с этим сейчас приходится моделировать их в виде программной реализации [4], что, конечно, отрицательно сказывается на производительности.

Таковой программной реализацией является sc-machine [5] – крайне эффективная имплементация обработки семантических сетей, написанная на языке программирования C++. Быстродействие достигается во многом за счет использования именно этого языка, так как он предоставляет программисту большой контроль над низкоуровневыми процессами, в том числе распределением и работой с памятью.

Существует большое количество агентов, которые и занимаются обработкой знаний, находящихся в базе.

Из-за некоторых моментов реализации у текущей версии есть ряд проблем, связанных с созданием масштабируемых, распределенных систем, поэтому предлагается другой подход к хранению и обработке этой информации.

III. ПРЕДЛАГАЕМЫЙ ПОДХОД

Как альтернативу предлагается использовать графовую систему управления базами данных (далее – СУБД) в качестве хранилища и событийно-ориентированный (event-driven) подход для обработки графов, а соответственно и знаний, которые представлены в виде семантических сетей.

Использование графовой СУБД уже дает преимущество во взаимодействии со знаниями, так как представление данных в базе, запросы в базу принимают «графовый вид», чем, по сути, и является семантическая сеть.

При этом одним из вариантов применения событийно-ориентированного подхода является создание серверных приложений в случае, когда нежелательно порождение лишних обслуживающих процессов. В многоагентной системе, к каковым относятся системы, разработанные на основе технологии OSTIS, происходит много вызовов различных агентов, они могут взаимодействовать друг с другом, что порождает дополнительные вызовы процедур. Таким образом, обработку всех этих вызовов необходимо вынести за рамки основного приложения, на что как раз таки в данном случае направлен событийно-ориентированный подход.

IV. ПРИНЦИПЫ РЕАЛИЗАЦИИ ПРЕДЛАГАЕМОГО ПОДХОДА

В качестве базы данных предлагается использовать Neo4j [6]. Neo4j – графовая СУБД с открытым исходным кодом, реализованная на Java. API для взаимодействия с Neo4j разработан для многих языков программирования, включая Python, Java, Go. Также программный интерфейс можно расширить с помощью серверных плагинов. К таковым, например, относится «Neo4j Streams Kafka Integration», который будет рассмотрен более подробно далее.

Для обмена сообщениями, событиями предлагается использовать Apache Kafka [7] – платформу распределенной потоковой передачи событий с открытым исходным кодом, которая используется тысячами компаний.

Принципиальное отличие Apache Kafka от других очередей задач и им подобным (AWS SQS, RabbitMQ) состоит в том, что сообщения (события) в очереди остаются после их обработки, что позволяет взаимодействовать с ними нескольким разным обработчикам. В отношении

многоагентных систем это отличие является довольно значительным в связи с тем, что одно и то же событие иногда должно быть обработано разными агентами.

Это же отличие позволяет обеспечить высокую степень отказоустойчивости и легкость восстановления после сбоев: Вам не придется после сбоя записывать все сообщения в очередь заново для их обработки, потому что они все еще находятся в очереди.

Как упоминалось ранее, для Neo4j есть плагин для интеграции с Apache Kafka – «Neo4j Streams Kafka Integration». Поток выполнения выглядит следующим образом:

1. основное приложение делает вызов удаленной процедуры с определенными параметрами;
2. обработчик видит этот вызов и начинает выполнять задачу;
3. по окончании выполнения обработчик сообщает о том, что результат готов и его необходимо «поднять» в основную базу;
4. основное приложение видит готовность задачи и обновляет данные в базе, используя результат выполнения задачи обработчиком.

V. ЗАКЛЮЧЕНИЕ

В работе рассмотрен подход к решению проблемы создания легко масштабируемых, распределенных систем на базе технологии OSTIS благодаря использованию графовых СУБД и событийно-ориентированного подхода.

1. W3C. Semantic Web – Mode of access: <https://www.w3.org/standards/semanticweb/>. – Date of access: 18.10.2021.
2. Голенков, В.В. Онтологическое проектирование гибридных семантически совместимых интеллектуальных систем на основе смыслового представления знаний / В.В. Голенков, Н.А. Гулякина, И.Т. Давыденко, Д.В. Шункевич, А.П. Еремеев // Онтология проектирования. – 2019. – Т. 9, № 31. – С. 132-151. – DOI: 10.18287/2223-9537-2019-9-1-132-151.
3. Семантические технологии проектирования интеллектуальных систем и семантические ассоциативные компьютеры / В.В. Голенков [и др.] // Доклады БГУ-ИП. – 2019. – № 3 (121). – С. 42-50.
4. Shunkevich, D. Ontological approach to the development of a software model of a semantic computer based on the traditional computer architecture / D. Shunkevich, D. Koronchik // Open Semantic Technologies for Intelligent Systems (OSTIS-2021): сборник научных трудов / Белорусский государственный университет информатики и радиоэлектроники; редкол.: В.В. Голенков [и др.]. – Минск, 2021. – Вып. 5. – С. 75-92.
5. sc-machine – Mode of access: <http://ostis-dev.github.io/sc-machine/>. – Date of access: 18.10.2021.
6. Neo4j – Mode of access: <https://neo4j.com/>. – Date of access: 18.10.2021.
7. Apache Kafka – Mode of access: <https://kafka.apache.org/>. – Date of access: 18.10.2021.