Хранение семантических сетей

Корончик Д. Н.
Кафедра интеллектуальных информационных технологий
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
e-mail: denis.koronchik@gmail.com

***Аннотация*—В этой статье описывается архитектура информационной подсистемы обеспечивающей хранение семантических сетей и ассоциативный доступ к ним.**

Ключевые слова: семантические сети; хранение и обработка семантических сетей; программная реализация хранилища семантических сетей

I. Введение

Концепция и технологии Semantic Web – это динамично развивающиеся направления интеллектуальных информационных технологий [1], которые обеспечивают возможность совместного многократного использования знаний различными приложениями, организациями и сообществами, позволяя компьютерам обрабатывать информацию на семантическом уровне.

Ключевым элементом информационных систем (ИС), основанных на технологиях Semantic Web, являются базы знаний. При создании крупных информационных систем по мере роста объемов используемых онтологий их хранение в линейном виде оказывается непродуктивным. В связи с этим актуальна проблема организации эффективных хранилищ баз знаний.

В рамках проекта OSTIS [2] ведутся работы по реализации sc-хранилища. Под sc-хранилищем понимается информационная подсистема, предназначенная для хранения sc-текстов (тексты, записанные с помощью SC-кода [3], любой sc-текст представляет собой семантическую сеть). Основными функция sc-хранилища являются:

- организация хранения текстов записанных с помощью SC-кода;
- предоставление программного интерфейса для добавления, удаления и извлечения хранимой информации (ассоциативный доступ);
- поддержка функций администрирования хранимой информации (распределение прав доступа);
- поддержка асинхронного выполнения запросов через программный интерфейс.

Эффективная реализация sc-хранилища должна удовлетворять следующим требованиям:

- высокая производительность – минимизация времени затрачиваемого на выполнение операций добавления, удаления и доступа к хранимой информации;
- минимальные затраты памяти и дискового пространства для хранения sc-текстов;
- масштабируемость – возможность простого добавления вычислительных мощностей при увеличении нагрузки.

II. Архитектура sc-хранилища

Важным при проектировании sc-хранилища является выбор способа адресации хранимых в ней sc-элементов [3]. От того каким будет адрес sc-элемента зависит многое: максимальное количество sc-элементов, которые могут быть помещены в хранилище; скорость поиска необходимого sc-элемента по его адресу и т. д. Для адресации sc-элементов в рамках хранилища был выбран подход сегментной адресации (адрес состоит из двух частей: сегмент и смещение). При длине адреса sc-элемента в 4 байта, максимальное число sc-элементов, которые можно хранить в sc-хранилище равно 2^{32} (4294967296). На сегмент отводится 2 байта, и на смещение 2 байта. Таким образом в каждом сегменте sc-хранилища может находится 2^{16} sc-элементов и столько же может быть сегментов. В перспективе количество sc-элементов, которые могут храниться в sc-хранилище может быть увеличено путем расширения адреса. К примеру, если рассматривать распределенное sc-хранилище (хранение на большом количестве серверов), то в адрес sc-элемента добавляется еще и адрес сервера на котором он хранится (4 байта), что приведет к увеличению максимального числа sc-элементов до 2^{64}.

Как правило, количество доступной оперативной памяти на компьютере гораздо меньше чем дискового пространства. Очевидно, что даже если каждый sc-элемент будет занимать 1 байт в оперативной памяти, что конечно же невозможно, то объем всех данных в хранилище может достигнуть 4 Гб. Таким образом увеличение размера одного sc-элемента всего на байт приводит к увеличению необходимой оперативной памяти на 4 Гб (для полного хранилища). По этой причине необходимо экономить каждый байт при описании sc-элемента. Но тут есть и обратная сторона медали, чтобы сделать систему масштабируемой, необходимо чтобы все сегменты хранящие sc-элементы занимали одинаковый объем оперативной памяти. Для этого было решено использовать для хранения различных sc-элементов структуру с фиксированным размером.

В SC-коде, который используется для представления информации в sc-хранилище можно выделить 3 типа sc-элементов: sc-узел, sc-ссылка, sc-коннектор. Для представления каждого из этих трех типов sc-элементов необходимы разные поля, поэтому они имеют разный размер. Поэтому каждый sc-элемент в сегменте имеет размер равный максимальному размеру для выше перечисленных типов (таковым является sc-коннектор).

Одной из сложностей с которой пришлось столкнуться – это каким образом хранить инцидентность между различными sc-элементами

(начальные и конечные sc-элементы для sc-коннекторов). Проблема заключается в том, что априори не известно количество присоединенных к sc-элементу коннекторов, что не дает возможности использовать списки инцидентности с фиксированным размером. Решить проблему удалось поместив эти списки неявно внутрь хранимых sc-коннекторов. Каждый sc-элемент имеет поле в котором хранится адрес sc-элемента, который является первой выходящей sc-дугой (в списке инцидентности) из этого элемента, а также поле, которое хранит адрес sc-элемента, который является первой входящей дугой. Каждая sc-дуга (sc-коннектор) находится в двух списках смежности: список входящих дуг для элемента который является её окончанием и список выходящих дуг для элемента который является её началом. Основываясь на этом в sc-элементы, которые обозначают sc-коннекторы были добавлены два дополнительных поля. Одно из этих полей указывает на следующий sc-коннектор из списка входящих дуг, а второе поле указывает на следующий элемент из списка выходящих дуг (Рис 1.).

III. Заключение

Описанные выше приемы позволили реализовать прототип sc-хранилища в котором есть ряд преимуществ:

- используемый подход адресации sc-элементов позволяет загружать сегменты с диска в память, а также выгружать их обратно на диск в любой момент и при этом данная операция не требует дополнительных преобразований. Все содержимое из оперативной памяти без изменений попадает на диск. Это дает возможность выгружать не используемые сегменты на диск, что позволяет sc-хранилищу абстрагироваться от имеющихся ресурсов оперативной памяти и работать на любых её объемах;
- максимальное количество хранимых sc-элементов можно увеличивать путем расширения адреса sc-элемента;
- сегментная адресация позволяет достаточно легко организовать многопоточную обработку хранимой информации.

Полученный прототип имеет следующие характеристики:

- размер sc-элемента 52 байта, таким образом размер сегмента 3.25 Мб. Размер полного sc-хранилища 208 Гб. Как видно потребуется 208 Гб места для хранения на жестком диске, что является вполне приемлемым;
- скорость заполнения sc-хранилища sc-узлами - 5116188.644108 узлов/сек;
- скорость заполнения sc-хранилища sc-дугами - 1833769.525978 sc-дуг/сек;
- скорость загрузки сегментов с диска в память – 305.56 сегмент/сек;
- скорость выгрузки сегментов с памяти на диск – 12 сегмент/сек;

Полученные данные отражают работоспособность sc-хранилища в однопоточном варианте исполнения, в качестве тестовой машины использовался ноутбук с 4 Гб оперативной памяти, процессором Intel Core i3 2.27 ГГц. Тесты проводились на операционной системе Ubuntu 12.04 32-bit.

[1] Пузанков Д.В. [и др.]. Интеллектуальные агенты, многоагентные системы и семантический Web: концепции, технологии, приложения. СПб: Изд-во «Технолит», 2008. 292 с.

[2] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2011. - Режим доступа: http://ostis.net. – Дата доступа: 10.09.2012

[3] Представление и обработка знаний в графодинамических ассоциативных машинах /В. В. Голенков, [и др]; – Мн. : БГУИР, 2001

[4] Программирование в ассоциативных машинах / В.В. Голенков [и др.]. – Минск, БГУИР, 2001 – 276 с.

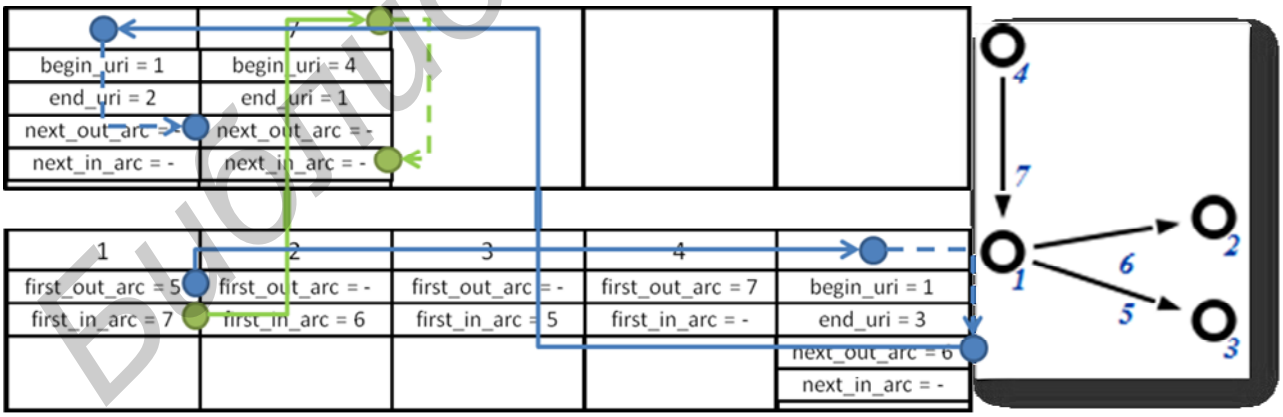

Рис. 1. Схема представления списков смежности в sc-хранилище