

Семантические данные и хранилища семантических данных

Содержание

1.	Семантические данные	2
1.1	Краткий экскурс	2
1.2	Семантика, Наука о Значении (Смысле).....	3
1.3	Логика	3
1.4	Семантические Веб-Языки	4
1.5	Вавилонская Башня	6
1.6	Семантическая сеть в виде базы данных	7
2.	Семантическая сеть в виде базы данных	7
2.1	Семантическая сеть в виде базы данных	8
2.2	Программные платформы для хранения семантических данных	9
3.	Онтология	10
3.1	Онтология в философии.....	10
3.2	Онтология в информатике	11
3.3	Семантические технологии на BBC.....	12
3.4	Семантические технологии в онтологии BBC – Sport.....	13
4.	База знаний	13
5.	NoSQL база данных.....	14
5.1	Теорема CAP.....	15
	Согласованность (Consistency)	15
	Доступность (Availability)	15
	Устойчивость к разделению (Partition tolerance).....	15
5.2	NoSQL базы данных.....	16

1. Семантические данные

Семантическая сеть – это расширение Мировой Сети через стандарты, установленные Консорциумом мировой сети.

Цель семантической сети - сделать данные Интернета машиночитаемыми.

Этот термин был придуман Тимом Бернерсом-Ли для обозначения сети данных, которые могут обрабатываться машинами, то есть такой, в которой большая часть смыслового значения является машиночитаемой.

Бернерс-Ли первоначально выразил свое видение Семантической паутины в 1999 году следующим образом:

У меня есть мечта о сети [в которой компьютеры] смогут анализировать все данные в сети - контент, ссылки и транзакции между людьми и компьютерами. «Семантическая сеть», которая делает это возможным, еще не появилась, но, когда она появится, повседневные механизмы торговли, бюрократии и нашей повседневной жизни будут управляться машинами, общающимися с машинами. "Интеллектуальные агенты", которых люди веками рекламировали, наконец-то материализуются.

Бернерс-Ли в настоящее время является директором Консорциума Всемирной паутины (W3C), который курирует разработку стандартов семантической паутины. С 2013 года деятельность семантической сети была отнесена к деятельности Сети Данных.

1.1 Краткий экскурс

С самого начала CS занималась обработкой данных. Языки программирования предоставляют простые и сложные типы данных для хранения самих данных. Первоначально семантика этих данных была встроена в программы, в которых они интерпретировались и использовались.

Около 50 лет назад данные начали отделяться от прикладной программы для хранения в базах данных. Это позволяло повторно использовать одни и те же данные в разных контекстах программирования и предотвращало повторную реализацию одного и того же компонента управления данными во многих приложениях. Тот факт, что смысл данных больше не был напрямую встроен в прикладную программу, привел к созданию механизмов для представления структуры и семантики разрабатываемых данных.

Одной из таких чрезвычайно успешных структур была реляционная модель данных.

Третья область вычислительной семантики была основана примерно в 1955 году с целью позволить компьютеру действовать разумно, как люди, то есть генерировать Искусственный Интеллект. Эта область началась с реализации общих методов решения проблем, таких как глобальный поиск и доказательство теорем. Однако, по прошествии короткого промежутка времени численная сложность задач, связанных с интеллектуальным

решением проблем, сделала очевидным, что требуется машинно-понятное представление знаний, связанных с тем, как проблема может быть решена эффективно.

1.2 Семантика, Наука о Значении (Смысле)

В реляционной базе данных все представлено в виде таблицы, у строки есть ключ, а у столбца есть имя. Благодаря этому, даже с помощью очень простой машины, можно найти номер телефона мистера X, если X - значение столбца имени, а номер телефона - заголовок другого столбца.

К сожалению, со средней веб-страницей это намного сложнее. Как упоминалось ранее, в различных HTML-тегах скрыто имя и где-то еще номер телефона (набор целых чисел, включая некоторые специальные символы).

Браузер необходим для отображения информации, а человек-читатель - для понимания информации на основе макета веб-сайта. Это решение в том виде, в каком оно было реализовано в сети, которое было представлено 20 лет назад. Как отмечалось ранее, чистая простота сделала Интернет невероятным успехом, который сейчас насчитывает более одного миллиарда пользователей. Его простота также оставляет пространство для улучшения.

Семантическая технология добавляет теги к полуструктурированной информации, как база данных добавляет заголовки столбцов к табличной информации. Воспользуемся небольшим примером на рис.1.

```
<person>
  <name>Sir Tim</name>
  <phone number>01-444444</phone number>
</person>
```

Рисунок 1 - Маленький пример

Эти аннотации позволяют компьютеру «понять», что сэр Тим — это имя человека, а 01-444444 - его номер телефона. Аналогичным образом программы и другие вычислительные ресурсы могут быть описаны с помощью семантических аннотаций. В этом суть технологии семантической паутины.

Из этого примера видно, что для определения семантики информации необходимы две вещи: язык вида $\langle X \rangle Y \langle /X \rangle$, для определения значения Y и такие термины, как X, для обозначения этого значения.

1.3 Логика

С алгоритмической точки зрения реализация систем логического мышления ясно демонстрирует, насколько сложной разрешимостью и сложностью является управление. Далее приведены логические парадигмы возрастающего уровня сложности.

- Логика высказываний — это довольно простой логический язык, обеспечивающий такие предложения, как A, B, C, ... и логические

связки, такие как AND и OR. Все интерпретации — это просто перечисление всех возможных ложных и истинных присвоений этим предложениям. Следовательно, логика высказываний разрешима, хотя уже у нее сложность NP.

- Логика предикатов первого порядка предоставляет более богатые средства для определения таких предложений, предоставляя такие термины, как c , $f(c, X)$, ... и предикатные символы, которые могут применяться к этим терминам $P(c)$, $Q(c)$, $f(c, X)$, В терминах могут использоваться переменные, которые могут быть экзистенциально или все количественно выражены (т.е. либо должен существовать термин, отвечающий формуле, либо все термины должны соответствовать формуле). Логика предикатов первого порядка все еще полурешима.
- Логика предикатов второго порядка и сопоставимые языки снимают это ограничение. Здесь можно применять предикаты к другим предикатам или целым формулам и интерпретировать переменные как наборы, а не как отдельные элементы области интерпретации. К сожалению, для этих языков уже унификация, то есть вопрос о возможности замены двух терминов, является полурешимым, что означает, что в этих языках нет даже подхода для реализации логического вывода.
- Логика Описания предоставляет целое семейство подязыков логики первого порядка разной сложности. Распространенным среди этих языков является ограничение формализма унарными и двоичными предикатами (понятиями и свойствами) и ограничение использования функциональных символов и логических соединителей для построения сложных формул. Различные уровни сложности и разрешимость этих языков вытекают из точного определения этих ограничений.

Поэтому было определено и реализовано множество различных языков, многие из которых содержат труднорешимое поведение наихудшего случая, но которые, тем не менее, все еще работают для многих практических приложений.

1.4 Семантические Веб-Языки

HTML предоставляет несколько способов выражения семантики данных. Очевидным является тег META.

До более широкого использования платформ описания ресурсов (RDF) системы использовали атрибут тега привязки для кодирования семантической информации. Также можно косвенно интерпретировать семантику HTML-документов. Например, информация, записанная в теге заголовка первого уровня ($\langle H1 \rangle$), может использоваться для кодирования концепций, которые существенно важны для описания содержания документа. Тем не менее, HTML не был разработан для предоставления описаний документов, помимо информирования браузера о том, как отображать содержимое. В рамках

усилий по расширению использования HTML для включения смысла был создан термин семантический HTML.

Расширяемый язык разметки (XML) был разработан как общий способ структурирования документов в Интернете. Он обобщает HTML, позволяя создавать теги, определяемые пользователем. Однако такая гибкость XML снижает возможности семантической интерпретации, которая была возможна с помощью предопределенных тегов HTML.

Платформа описания ресурсов (RDF) — это простая модель данных для семантического описания ресурсов в Интернете. Бинарные свойства связывают термины, образуя ориентированный граф. Эти термины, а также свойства описываются с помощью универсального идентификатора ресурса (URI).

Поскольку свойство может быть URI, его снова можно использовать как термин, связанный с другим свойством. То есть, в отличие от большинства логических языков или баз данных, невозможно отличить язык или схему от операторов языка или схемы.

Например, в операторе `<rdf: type, rdf: type, rdf: Property>` указано, что тип относится к свойству типа.

Кроме того, в отличие от обычного гипертекста, в RDF URI могут относиться к любому идентифицируемому объекту (например, человеку, транспортному средству, бизнесу или событию). Эта очень гибкая модель данных, очевидно, подходит в контексте свободного и открытого Интернета; однако это создает большую головную боль для логиков, желающих наложить новый язык поверх других.

Схема RDF (RDFS) использует базовые операторы RDF и определяет простой язык онтологий. В частности, он определяет такие сущности, как `rdfs: class`, `rdfs: subclass`, `rdfs: subproperty`, `rdfs: domain` и `rdfs: range`, что позволяет моделировать классы, свойства с ограничениями домена и диапазона, а также иерархии классов и свойств. RDFS — это специальный словарь RDF для этой цели, который представляет собой просто RDF плюс еще несколько определений (операторов) в RDF.

Язык веб-онтологий OWL расширяет этот словарь до полного спектра логик описаний, определенных в RDF, а именно OWL Lite, OWL DL и OWL Full.

Предусмотрены механизмы для определения таких свойств, как обратных, переходных, симметричных или функциональных. Свойства могут использоваться для определения принадлежности экземпляров к классам или иерархий классов и свойств.

Откровенно говоря, OWL Lite уже представляет собой довольно выразительную логику описания, которая делает разработку эффективных реализаций для больших наборов данных довольно сложной, а на практике такой же сложной, как реализация OWL DL.

Однако ни один из этих языков не может использовать полный RDF, т.е. некоторые допустимые операторы RDF недействительны в Lite или DL. Это

связано с тем, что языки логики, такие как логика описаний, исключают мета-утверждения, то есть утверждения вместо утверждений.

Для RDF и RDFS это не было проблемой, поскольку ни один из языков не предоставлял механизмов для определения сложных логических определений.

В двух словах, Lite и DL определяют словарь в RDF и ограничивают использование RDF. OWL Full снимает эти ограничения. OWL Full предоставляет словарь OWL DL, то есть выразительную логику описания, и позволяет использовать любой допустимый оператор RDF. Например, в OWL Full класс можно рассматривать одновременно как набор индивидов и как индивид. Следовательно, OWL Full выходит за пределы выразительной области логики описания и минимально требует вывода типа средства доказательства теорем, такого как логика первого порядка (т.е. является полуразрешимым).

1.5 Вавилонская Башня

Модель взаимодействия открытых систем (модель OSI) является продуктом усилий Международной организации по стандартизации в области взаимодействия открытых систем. Это способ разделения системы связи на более мелкие части, называемые уровнями. Уровень — это набор концептуально похожих функций, которые предоставляют услуги вышележащему слою и получают сервисы от нижележащего уровня.

Эта модель широко используется при проектировании сетевых архитектур в глобальном масштабе. Модель начинается с физического уровня и заканчивается прикладным уровнем, который предоставляет такие механизмы, как протокол HTTP.

Например, в стеке Интернета компоненты протокола Интернета IP и TCP находятся на уровнях 3 и 4.

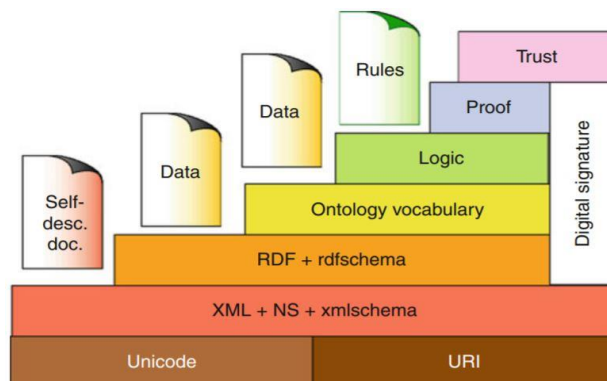


Рисунок 2 - Вавилонская Башня, структурирующая Семантическую Сеть

Тим Бернерс-Ли начал аналогичную концептуальную попытку структурировать семантическую сеть, как показано на рисунке 2.

На самом низком уровне Unicode рассматривается как средство кодирования текста, URI для ссылки на ресурсы и XML с его пространством имен и механизмами схемы для обеспечения синтаксических описаний структурированных объектов.

Вдобавок к этому он предусмотрел пять уровней семантики: RDF, OWL, RIF и уровни доказательства и доверия.

Этот тип разделения на уровни выполняет две основные функции: предотвращение повторной реализации функциональных возможностей нижнего уровня на верхнем уровне и разрешение приложению, которое понимает только нижний уровень, по крайней мере, интерпретировать части определений на более высоком уровне.

1.6 Семантическая сеть в виде базы данных

Семантическая сеть Интернет как область исследований стала свидетелем объединения ряда сообществ, включая искусственный интеллект (от агентов, моделирования знаний и логики) и Интернет. Однако по большей части совпадение исследований Семантической Сети и баз данных было минимальным. Это может показаться несколько неожиданным, так как в настоящее время Сеть данных является широко используемым термином, но в первые дни акцент делался на создании структур знаний в качестве платформы для агентов.

Как мы можем использовать его как базу данных? Основные проблемы исследования, которые в настоящее время начинают возникать, включают следующее:

Какие конкретные методы базы данных (например, многораздельные хэши, таблицы столбцов) наиболее применимы к высокопроизводительному хранилищу RDF?

Как структурировать тесты для крупномасштабных репозиториях? В том числе какие размеры правильные?

Когда и где следует обращаться к рассуждениям? Например, материализация (предварительное вычисление и сохранение предполагаемых троек) - дорогостоящий процесс, который может не способствовать достижению желаемых результатов.

2. Семантическая сеть в виде базы данных

Хранение и извлечение информации являются важными функциями информационных систем (ИС). Эти функции были реализованы в течение десятилетий благодаря завершенности технологии реляционных баз данных. В последние годы появилась концепция информационной системы (SIS), в которой информация представлена с явной семантикой, основанной на ее

значении, а не на синтаксисе, чтобы обеспечить ее автоматическую и интеллектуальную обработку компьютерами.

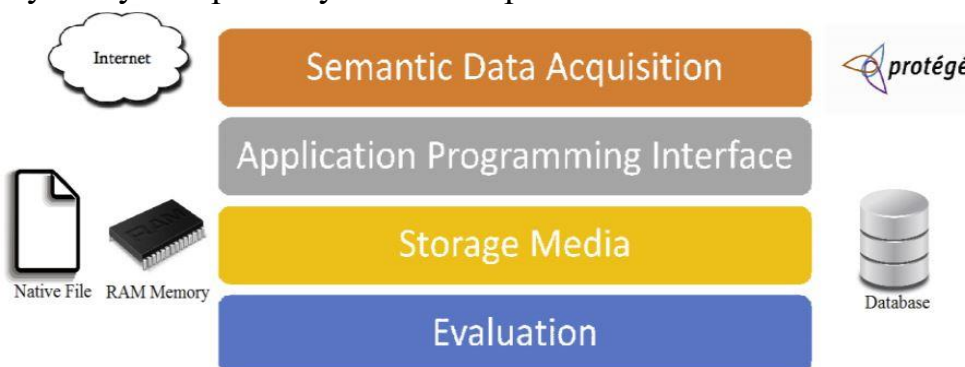


Рисунок 3 - Структура, разработанная на этапе предложения метода проектных исследований

На рис. 3. представлена структура, разработанная на этапе предложения метода проектных исследований для анализа подходов к семантическому хранению данных.

- Первый уровень структуры — это уровень сбора семантических данных. Цель этого слоя — получить онтологии или семантические данные, которые будут использоваться другими слоями.
- Уровень сбора семантических данных Фреймворка на рисунке использует существующие онтологии, которые были разработаны и общедоступны в Интернете.
- Второй уровень — это уровень интерфейса прикладного программирования (API); он используется для создания, редактирования, просмотра и удаления онтологий или семантических данных. Он также используется для загрузки существующих семантических данных.
- Третий слой, который является носителем информации, используется для физического хранения семантических данных в памяти компьютера.
- Оценка — это последний уровень; он анализирует и обсуждает базовые структуры, используемые для хранения семантических данных.

2.1 Семантическая сеть в виде базы данных

Для хранения онтологических или семантических данных используются три способа, а именно: в памяти, в собственных или файловых системах и базах данных.

При способе хранения в памяти центральная память компьютера используется для хранения семантических данных.

Преимущество этого способа заключается в том, что он обеспечивает быстрое время отклика на запросы с помощью семантических данных малого масштаба.

Основные недостатки этого способа заключаются в том, что большие семантические данные трудно обрабатывать, а сохраненные данные не хранятся постоянно. Фактически, при таком подходе семантические данные необходимо загружать в память компьютера по требованию, что неэффективно и отнимает много времени.

Способ хранения в собственных или файловых системах использует файлы для хранения семантических данных; это обеспечивает быструю загрузку и запрос семантических данных. Обработка крупномасштабных семантических данных является одним из главных недостатков этого способа.

Кроме того, такие функции, как оптимизация запросов, восстановление данных, обработка транзакций и контролируемый доступ, должны быть реализованы отдельно; к счастью, эти недостатки устраняются с помощью способа хранения в базах данных. Фактически, реляционные базы данных (RDB) остаются подходящим носителем для хранения семантических данных из-за завершенности технологии реляционных баз данных.

Хранилище семантических данных в базе данных предлагает множество функций, включая хранение, запросы, логику и масштабируемость. Для хранения статистических данных в базах данных используются два способа: общая и конкретная схемы.

При использовании общей схемы таблица используется для хранения семантических данных в базе данных RDB; столбцы таблицы являются элементами инструкций RDF онтологии. Улучшенная версия способа использования общей схемы называется нормализованным тройным хранилищем; она использует еще две таблицы для хранения семантических данных с целью удешевления запросов на объединение.

2.2 Программные платформы для хранения семантических данных

Для обеспечения возможности хранения и запроса семантических данных было разработано несколько платформ. Наиболее популярными из этих платформ являются: AllegroGraph, Jena, OpenAnzo, Minerva и Sesame (рис. 4.).

Platform	License	Operating system	Type of Storage
AllegroGraph	Commercial/Free	Linux	Native
Jena	Free/Open Source	Windows/Linux	Memory, Native, RDB
Sesame	Free/ Open Source	Windows/Linux	Memory, Native, RDB
Open Anzo	Free/ Open Source	Windows Linux	RDB
Minerva	Free	Windows/Linux	RDB

Рисунок 4 - Программные платформы для хранения семантических данных

Allegrograph — это серверное приложение, к которому клиентские приложения получают удаленный доступ. Он позволяет хранить и запрашивать семантические данные и предоставляет API для прямого доступа к этим данным без какого-либо использования запросов.

Minerva является компонентом Интегрированного инструментария разработки онтологий; он используется в качестве библиотеки в интегрированной среде разработки Eclipse (IDE) для хранения семантических данных.

Open Anzo был разработан IBM; он может использоваться в трех различных режимах для хранения и запроса семантических данных: (1) встроен в приложение, (2) установлен в качестве серверного приложения и доступен удаленно клиентам или (3) запущен локально.

API приложения Jena интегрирован в Eclipse IDE в виде библиотеки; он позволяет создавать и хранить семантические данные в различных форматах.

Сезам — это набор для разработки программного обеспечения (SDK), который был разработан в рамках европейского проекта IST On-to-Knowledge. Это позволяет запрашивать или экспортировать семантические данные.

3. Онтология

3.1 Онтология в философии

Онтология — это раздел философии, изучающий такие понятия, как существование, бытие, становление и реальность, как показано на рис. 5.

Aristotle's ontology:

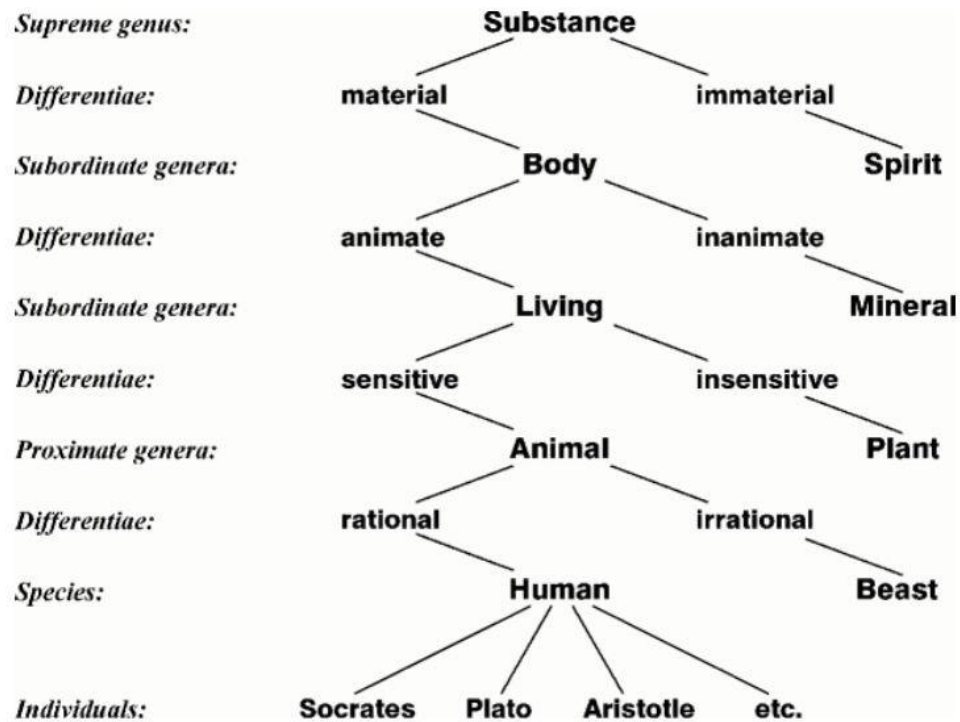


Рисунок 5 - Онтология Аристотеля

Онтология включает в себя вопросы о том, как сущности группируются в основные категории и какие из этих сущностей существуют на самом фундаментальном уровне. Онтологию иногда называют наукой о бытии и она относится к основной отрасли философии - метафизике.

3.2 Онтология в информатике

В информатике онтология — это формальное описание знаний как набора понятий в рамках предметной области и отношений, которые существуют между ними. Чтобы дать такое описание, нам необходимо формально указать такие компоненты, как индивиды (экземпляры объектов), классы, атрибуты и отношения, а также ограничения, правила и аксиомы. В результате онтологии не только предоставляют общее и многократно используемое представление знаний, но также могут добавлять новые знания о предметной области.

Конечно, существуют и другие методы, использующие формальные спецификации для представления знаний, такие как словари, классификаторы, энциклопедии, тематические карты и логические модели. Однако, в отличие, например, от классификаторов или схем реляционных баз данных, онтологии выражают взаимосвязи и позволяют пользователям различными способами связывать несколько концепций с другими концепциями.

Поскольку онтологии определяют термины, используемые для описания и представления области знаний, они используются во многих приложениях для определения взаимосвязей и повышения эффективности управления знаниями.

Принятие онтологий помогает раннему тестированию гипотез в фармацевтике путем классификации выявленных явных связей с онтологией причинно-следственных связей. Онтологии также обогащают семантический веб-майнинг, анализ данных о состоянии здоровья для анализа, обнаружения мошенничества и семантической публикации.

3.3 Семантические технологии на BBC

В качестве примера применения онтологии можно привести успешную онтологию BBC.

Веб-служба BBC является одним из самых посещаемых веб-сайтов и крупнейшим в мире новостным сайтом. По состоянию на 2007 год он содержал более двух миллионов страниц.

Основное внимание было уделено отдельным, автономным HTML-микросайтам, которые не связаны друг с другом, и другим источникам данных в Интернете. Трудно найти все, что BBC опубликовала о каком-либо конкретном объекте. Мы не можем перейти со страницы о музыканте на страницу со всеми программами, в которых играл этот артист, к их биографии и т.д.

Используя онтологии и готовую для предприятия базу знаний с возможностью вывода, BBC хотела свести к минимуму дорогостоящее редакционное управление ресурсами контента. Они также хотели, чтобы навигация по веб-сайту основывалась на том, что важно для потребителя (например, команды, страны, игроки и т.д.). В результате BBC ожидала увидеть увеличение объема агрегирования контента, повторного использования и переориентации без дополнительных затрат.

Таким образом, в BBC используется метод онтологии, этот сайт предоставляет доступ к технологиям, которые использует BBC для поддержки своей аудитории, обращающейся к таким приложениям, как BBC Sport, BBC Education, BBC Music, новостные проекты и многое другое. Эти онтологии составляют основу нашей платформы связанных данных.

Онтологии строятся постепенно в соответствии с текущими бизнес-требованиями. Ожидается, что все они будут развиваться по мере развития наших требований. BBC выпускает множество богатого и разнообразного контента о вещах, которые важны для нашей аудитории. Связанные данные дают нам возможность объединить контент с помощью этих тем. Они используют онтологии для описания окружающего нас мира, контента,

создаваемого ВВС, а также управления, хранения и обмена этими данными с помощью платформы связанных данных.

3.4 Семантические технологии в онтологии ВВС – Sport.

Спортивная онтология — это простая облегченная онтология для публикации данных о спортивных соревнованиях. Термины в этой онтологии позволяют публиковать данные о структуре спортивных турниров как серии мероприятий, участии агентов в соревнованиях, типе дисциплины, в которой участвует мероприятие, награде, связанной с соревнованием, и как она была получена и т.д.

Несмотря на то, что она возникает в конкретном случае использования ВВС, спортивная онтология должна быть применима к широкому спектру вариантов публикации данных о спортивных соревнованиях. Была предпринята попытка обеспечить совместимость с более общими используемыми онтологиями. В частности, она в значительной степени опирается на онтологию события.

4. База знаний

База знаний — это совокупность взаимосвязанных описаний сущностей (объектов реального мира, событий, ситуаций или абстрактных понятий), взаимосвязанных таким образом, чтобы обеспечить хранение, анализ и повторное использование этих знаний с помощью машинной интерпретации.

Большинство людей знакомы с традиционными реляционными базами данных. Там есть ячейки и таблицы, заполненные буквами и цифрами. Годы усовершенствований и оптимизаций позволили организациям управлять феноменальными объемами данных. Но, как лучше всего сказал американский писатель Клиффорд Столл: данные — это не информация, информация — это не знание.

Базы знаний абстрагируются от простой базы данных, чтобы создать организованный набор данных, который ближе к тому, как человеческий мозг организует информацию. Базы знаний добавляют к данным семантическую модель, которая включает формальную классификацию с классами, подклассами, отношениями и экземплярами (онтологиями и словарями), с одной стороны, и правилами интерпретации данных, с другой.

Разница между базой данных и базой знаний заключается в том, что база данных представляет собой набор данных, представляющих факты в их базовой форме, в то время как база знаний хранит информацию в виде ответов на вопросы или решений проблем. База знаний обеспечивает быстрый поиск, извлечение и повторное использование. Информация в базе знаний, как правило, полностью разработана и готова к применению.

5. NoSQL база данных

База данных NoSQL предоставляет механизм хранения и извлечения данных, который моделируется иными способами, чем табличные отношения, используемые в реляционных базах данных.

База данных NoSQL — это нереляционная система управления данными, которая не требует фиксированной схемы. Она позволяет избежать соединений и легко масштабируется. Основная цель использования базы данных NoSQL — для распределенных хранилищ данных с огромными потребностями в хранении данных. NoSQL используется для больших данных и веб-приложений реального времени. Например, такие компании, как Twitter, Facebook и Google, собирают терабайты пользовательских данных каждый день.

Традиционная СУБД использует синтаксис SQL для хранения и извлечения данных для дальнейшего просмотра. Вместо этого система баз данных NoSQL охватывает широкий спектр технологий баз данных, которые могут хранить структурированные, полуструктурированные, неструктурированные и полиморфные данные.

Концепция баз данных NoSQL стала популярной среди интернет-гигантов, таких как Google, Facebook, Amazon и др., которые имеют дело с огромными объемами данных. Время отклика системы замедляется, когда вы используете СУБД для больших объемов данных.

Чтобы решить эту проблему, мы могли бы "расширить" наши системы, обновив наше существующее оборудование. Этот процесс является дорогостоящим, что показано на рис. 6.

Альтернативой этой проблеме является распределение нагрузки на базу данных на нескольких хостах при увеличении нагрузки. Этот метод известен как "масштабирование".

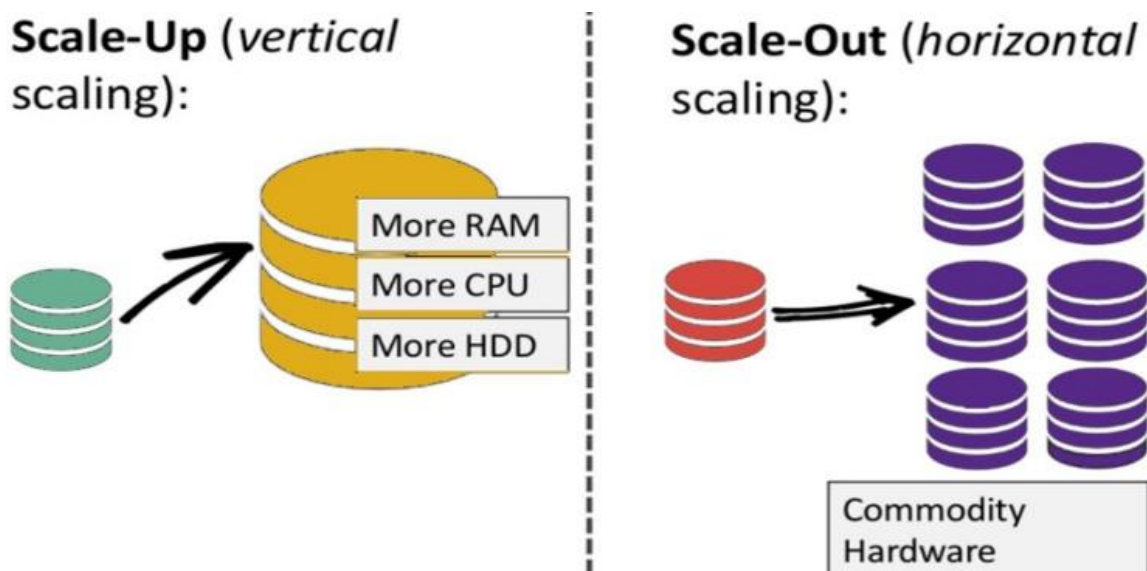


Рисунок 6 - Вертикальное и горизонтальное масштабирование

5.1 Теорема CAP

Теорема CAP также называется теоремой Брюера. В ней говорится, что распределенное хранилище данных не может предоставлять более двух из трех гарантий: согласованность, доступность, устойчивость к разделению.

Согласованность (Consistency)

Данные должны оставаться согласованными даже после выполнения операции. Это означает, что после записи данных любой будущий запрос на чтение должен содержать эти данные. Например, после обновления статуса заказа все клиенты должны иметь возможность видеть одни и те же данные.

Доступность (Availability)

База данных всегда должна быть доступной и отзывчивой. У него не должно быть никаких простоев.

Устойчивость к разделению (Partition tolerance)

Устойчивость к разделению означает, что система должна продолжать функционировать, даже если связь между серверами нестабильна. Например, серверы могут быть разделены на несколько групп, которые могут не взаимодействовать друг с другом. Здесь, если часть базы данных недоступна, другие части всегда остаются незатронутыми.

Классификация систем NoSQL как AP, CP или CA смутно отражает возможности отдельных систем и, следовательно, широко принята в качестве средства для сравнений высокого уровня. Однако важно отметить, что теорема CAP на самом деле ничего не говорит о нормальной работе; она просто говорит нам, способствует ли система доступности или согласованности перед лицом сетевого раздела.

5.2 NoSQL базы данных

Многие базы данных NoSQL были разработаны молодыми технологическими компаниями, такими как Google, Amazon, Yahoo и Facebook, чтобы обеспечить более эффективные способы хранения контента или обработки данных для огромных веб-сайтов. Некоторые из наиболее популярных баз данных NoSQL:

- Apache CouchDB, база данных с открытым исходным кодом, основанная на документах JSON, которая использует JavaScript в качестве языка запросов.
- Apache Cassandra, база данных хранилища с открытым исходным кодом и широкими столбцами, предназначенная для управления большими объемами данных на нескольких серверах и кластеризации, охватывающей несколько центров обработки данных.
- MongoDB, база данных на основе документов с открытым исходным кодом, которая использует документы и схемы, подобные JSON, и является компонентом базы данных стека MEAN
- Red — это мощное хранилище значений ключей в памяти, используемое для кэширования сеансов, очередей сообщений и других конкретных приложений.
- Elasticsearch — база данных на основе документов, включающая систему полнотекстового поиска.