Решение задачи о подписке в системе алгоритмической торговли

Шагал А.А. магистрант 2 курса кафедра Компьютерных технологий
Университет ИТМО shagal@rain.ifmo.ru

Аннотация

Решение задачи о подписке является базовым подходом при проектировании высоконагруженных распределенных систем. Подход позволяет уменьшить сетевой трафик между компонентами системы и реализовывать бизнес-логику взаимодействия различных компонентов. В данной работе рассматривается существующий язык подписок и событий, преимущества, а так же новый подход, позволяющий увеличить гибкость системы.

Введение

Решение задачи о подписке или система публикации/подписки– это подход, при котором в системе есть подписчики за информацией и источники информации. Подписчики с помощью заданного языка подписок формируют критерий отбора информации. Источники публикуют информацию. Задача системы публикации/подписки заключается в доставке информации только тем подписчикам, которые в ней заинтересованы. Системы публикации/-подписки часто применяются при проектировании систем алгоритмической торговли. В данной работе рассматривается определенная система и сервис финансовых инструментов, а так же их параметров. Система состоит из нескольких компонент, одна их которых отвечает за управление финансовыми инструментами и их параметрами. Остальные компоненты могут как модифицировать инструменты, так и подписываться на изменениями с помощью заданного языка подписок. На сервис финансовых инструментов и их параметров накладываются функциональные требования и требования к производительности. К функциональным требованиям относятся: гибкость системы и эффективность управления параметрами инструментов. Требования к производительности в системах алгоритмической торговли имеют большой приоритет. Важными мерами являются масштабирование, скорость и пропускная способность. Функциональные требования и требования к производительности зачастую противоречат друг другу. Чем больше функциональность системы – тем медленнее она работает. Задача данной работы – реализовать язык событий для сервиса финансовых инструментов в целях увеличения гибкости системы. В конце работы предлагается подход для реализации языка подписок.

Язык подписок и событий

Системы публикации/подписки имеют сложную классификацию относительно того, как подписчики формируют критерии, и как источники обновляют информацию и где развернута система. Первый два признака задаются языком подписок и событий. Выделяют два основных типа подписок: topic-based и content-based. Topic-based подписок изменение содержит определенный topic, далее тэг. Для рассматриваемой системы тэгом является уникальный идентификатор объекта. Примером сообщения с тэгом может быть $[IBM, bid = 239]$. Тэгом в данном случаи будет IBM, и сообщение попадет к подписчику заинтересованному в IBM. В content-based системах подписка задается логическим выражением использующем содержимое сообщения. Часто логическое выражение задается фильтром, где каждое условие – это огранение на значения аттрибута. Для примера с IBM, аттрибутом может быть bid, а условием на bid: $[bid \geq 239]$. Topic-based системы имеют меньшую мощность языка по сравнению с content-based. Существует несколько основных подход по реализации каждого из языков, при этом только topic-based системы дают производительность порядка 1-2 микросекунд на обработку простого события. Сервис финансовых инструментов поддерживает и topic-based язык и content-based язык.

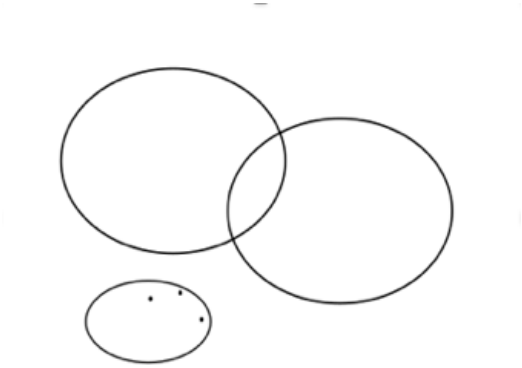

Рис. 1: Пространство подписок и событий

Далее тэг и уникальный идентификатор будут соответствовать одной и той же сущности. По языкам событий можно выделить два типа систем публикации/подписки. Простые обновления и поддержка иерархичной структу-

ры тэгов. Второй тип позволяет применять изменение ко множеству объектов. Данное свойство достигается за счет наследования изменений дочерними тэгами в иерархии. Сервис финансовых инструментов использует второй тип, добавляя к наследованию, переопределение. Вводится состояние объекта. Объект является точкой в пространстве. Пространство имеет размерность равную количеству атрибутов, использованных в content-based системах. Если для объекта определено значения в определенном измерении, то перед применением изменения необходимо проверить не является ли источник более приоритетным. Если является - изменение применяется. Если не

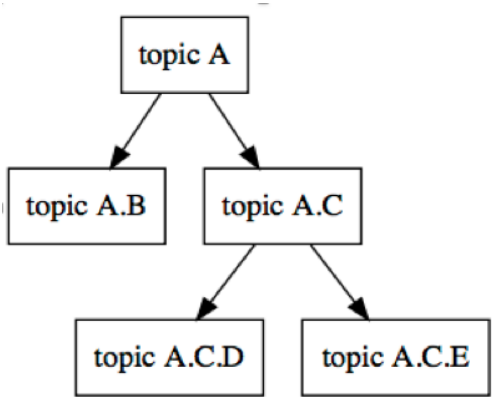

Рис. 2: Topic-based система с иерархией тэгов

является - изменение не активно для объекта. Данное свойство реализуется на базе иерархий тэгов и приоритетов иерархий. Основным приоритетом является высота тэга в иерархии. Далее считается, что тэгом в иерархии, не являющийся листом, называется группой. Листы иерархии называются инструментами.

Существующий язык событий обладает несколькими недостатками как с функциональной, так и с производительной точки зрения. К функциональным относятся:

- Группы не являются динамическими, то есть в случае, если бизнес логика клиента не соответствует иерархии групп, изменения проставляются на инструменты
- Множественное наследование приводит к неправильному использованию системы

Второй недостаток дает излишнюю гибкость системы. Первый недостаточную гибкость, а значит эффективность управления инструментами. К производительным недостаткам относятся сложности реализации наследований, являющиеся критическим местом.

Рассмотрим возможность реализации динамических групп. Для общего случая изменение множества инструментов задается фильтром. Фильтр задает динамическое множество инструментов, для которых активны изменения, заданные в сообщении. Далее необходимо отправить обновление всем подписчикам на инструменты, входящие в заданное множество. Простая реализация не поддерживает наследование и уточнение параметров инструментов, если оно необходимо. Обновление параметров на множество использует content-based механизм фильтрации. Для реализации наследования параметров введем следующее отношение для множеств: Множество A уточняется/-наследует параметры Множества B в том и только том случаи, если любой объект удовлетворяющий фильтру A, удовлетворяет фильтру B. Важно заметить, что множество всех объектов динамическое, а объект должен находиться в двух множествах на протяжении всего жизненного цикла. Чтобы рассчитать данное свойство предлагается проверять условие не имея информации о множестве, а базируясь только на фильтрах. В первую очередь необходимо перевести все фильтры в ДНФ, рассматривая каждую конъюнкцию отдельно. Фильтр в формате конъюнкции можно считать объектов и найти все фильтры, которые допускают данный. Важно заметить, что между фильтрами бывает три типа отношений.

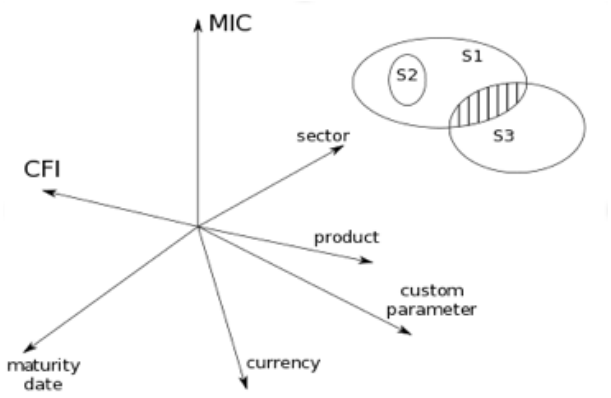

Рис. 3: Отношения между подписками

- Удовлетворяет: Фильтр A удовлетворяет фильтр B, если любой объект удовлетворяющий A удовлетворяет B. Например: $[A : bid > 239]$ и $[B : bid > 239 and CFI = OXXXX]$ Если объект удовлетворяет B, то и A.
- Заведомо не удовлетворяет: Фильтр A заведомо не удовлетворяет фильтр B, если любой объект удовлетворяющий A не удовлетворяет B. Например: $[A : bid > 239 and CFI = FXXXX]$ и $[B : bid > 239 and CFI = OXXXX]$ Если объект удовлетворяет B, то точно не удовлетворяет A.(CFI всегда будет противоречить)
- Независимы: Фильтр A независим от фильтра B, если тот факт, что объект удовлетворет A не дает информации о фильтре B. Например: $[A : bid > 239]$ и $[B : CFI = OXXXX]$

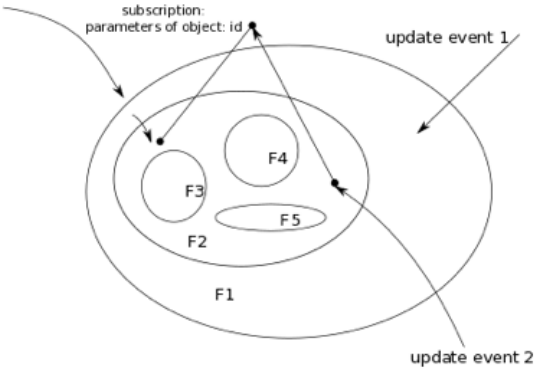

Рис. 4: Пример изменения объекта

Описанные три отношения дают возможность реализовать требуемую гибкость. Динамические группы реализуются за счет того, что в фильтре может быть любая информация о любых аттрибутах. Простейшие изменения по topic так же являются изменением на фильтр, где фильтром является текущее состояние объекта. В случаи если фильтра удовлетворяют друг друга – происходит переопределение изменений. Данные фильтры можно организовывать в иерархии. В случаи если два фильтра заведомо не удовлетворяют друг другу – происходит переопределение. Данные два фильтра являются источниками на разные множества, которые никогда не пересекутся.

Случай с независимыми фильтрами стоит рассматривать отдельно. Пусть есть два фильтра. Оба фильтра задают множества объектов на которые меня-

ется значения одного и того же измерения. Пусть данные множества пересеклись. Система не имеет возможности выбрать приоритетный источник. Но так же система не может выдать ошибку, так как ошибка произошла раньше, когда появился один из фильтров. Именно поэтому система должна выдавать ошибку при появлении независимого фильтра как только он появился. Появление такого фильтра является увеличивает вероятность того, что динамические множества заданные фильтром пересекутся. Таким образом обработка независимых фильтров дает требуемое ограничение гибкости системы.

Для реализации рассмотренного языка подписок необходимо использовать механизм агрегации подписок по источникам. Данный подход частично применяется в системе: SIENA. Появление очередной подписки означает, что необходимо найти фильтр являющийся актуальным источником для данной подписки. Организации фильтров в иерархии относительно заданных отношений дает возможность обхода в глубину этих иерархий. В случаи если подписка не может быть прокинута в дочерние фильтры активный фильтр найден. Важно заметить, что при открытии подписки необходимо знать состояние объекта. Изменение измерений множества будет занимать O(количество подписок, для которых фильтр является активным). Можно заметить, что это минимальное время, которого можно добиться при обработке событий.

Заключение

Предложен язык событий, решающий существующую проблему множественного наследования и недостаточной гибкости системы. Предложен подход к реализации описанного языка.

Литература

[1] H. K. Y. Leung. Subject space: A state-persistent model for publish/subscribe systems. In Proceedings of the 2002 conference of the Centre for Advanced Studies on Collaborative research, page 7. IBM Press, 2002

[2] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. ACM Transactions on Computer Systems, 19(3):332–383, Aug. 2001.

[3] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. ACM Computing Surveys (CSUR), 35(2):114–131, 2003.