

Распределённая система бухгалтерского учёта

Идея не новая. Если поискать в Интернет по словосочетанию distributed accounting, то можно найти много интересных статей и даже научные публикации по этому поводу в том числе, например, по реализации распределённого бухгалтерского учёта на основе технологии блокчейн.

Тем не менее темой данной статьи будет именно описание идеи реализации распределённой системы бухгалтерского учёта, в том числе как, например, развитие популярной системы 1С:Бухгалтерия.

Кроме этого, далее, для упрощения, речь будет вестись о бухгалтерском учёте, но на самом деле всё, что будет сказано ниже, можно отнести вообще к любому виду учёта: финансовому, управленческому и так далее.

Для начала я предлагаю забыть всё, что мы знаем о бухгалтерском учёте, его нюансах и регламентах. Это будет мешать восприятию дальнейшего "поточка мыслей". Давайте взглянем на бухгалтерский учёт "глазами ребёнка" так, как-будто мы его увидели в первый раз.

Итак, что такое бухгалтерский учёт? Это объекты учёта, например, товары, их характеристики (аналитика) и количественные показатели. Объекты учёта в процессе хозяйственной деятельности человека постоянно взаимодействуют между собой, порождая события учёта (проводки). При этом эти взаимодействия осуществляются по заранее определённым правилам, а не хаотично как им вздумается. Ещё одной важной составляющей бухгалтерского учёта является балансовый отчёт. Это по сути снимок состояния системы, всех объектов учёта, в какой-то фиксированный момент времени.

Таким образом модель бухгалтерского учёта в упрощённом абстрактном виде включает в себя или состоит из следующих сущностей:

- объекты учёта;
- характеристики объектов;
- количественные показатели;
- события взаимодействия;
- отчёт о состоянии на момент времени.

Теперь давайте себе представим, что мы играем в сетевую онлайн игру. Там есть гоблины, гномы, эльфы и прочие персонажи. Они бегают, дерутся и взаимодействуют друг с другом разными способами, но строго согласно правил и ограничений игры. При этом у них есть количественные характеристики в виде всевозможных навыков, уровня здоровья и прочее. Вот гном ударил эльфа дубиной: у одного уменьшилось здоровье, а у второго увеличился боевой опыт... В любой момент времени можно поставить игру на паузу и посмотреть состояние персонажей, какое место в рейтинге они занимают относительно друг друга в данный момент. Да, всё верно - модели сетевой игры и бухгалтерского учёта на верхнем абстрактном уровне идентичны. Что даёт нам это наблюдение ?

Технологии сетевых игр адаптированы для высоко нагруженных, отказоустойчивых, распределённых географически систем реального времени. При этом количественные характеристики персонажей постоянно меняются и пересчитываются. Вполне возможно, что, учитывая идентичность моделей бухгалтерского учёта и подобных игр, можно использовать игровые технологии для реализации распределённой системы учёта.

В качестве примера возьмём платформу для разработки игр, другими словами - игровой движок, [Microsoft Orleans](#). Точнее всё-таки будет сказать, что Orleans сначала был движком для игры Halo, а затем, в процессе своего развития, превратился в современную платформу для разработки распределённых приложений.

Цитата официального сайта Microsoft:

"Существуют, казалось бы, бесконечные способы использования Orleans. Некоторые из наиболее распространённых вариантов использования это создание следующих видов приложений: игры, банковские услуги онлайн, чат-приложения, GPS-трекинг, торговля акциями, корзины для покупок, приложения для голосования и многое другое. Orleans используется Майкрософт в Azure, Xbox, Skype, Halo, PlayFab, Gears of War и многих других внутренних службах."

Платформа основана на модели так называемых "акторов" – объектов, которые имеют состояние (характеристики) и поведение. Акторы взаимодействуют друг с другом при помощи сообщений. Кроме этого следует особенно отметить, что платформа поддерживает сохраняемость акторов, например, в базе данных, а также выполнение распределённых транзакций между ними.

Итак, судя по описанию и функциональности платформы Orleans, она вполне может быть использована для разработки распределённой системы бухгалтерского учёта. Актор – это ни что иное, как объект учёта или счёт учёта, в зависимости от того какую роль назначит ему разработчик системы. Характеристики или внутреннее состояние актора – это аналитика и количественные измерители. Поведение – правила учёта, а сообщения другим акторам – проводки. Если сюда ещё добавить акторов, которые будут играть роль показателей в отчётах, то вся картина складывается в единое целое.

Давайте для примера попробуем автоматизировать такую задачу как учёт складских остатков в режиме реального времени. Акторами, например, будут SKU. Приход товара – это сообщение соответствующим акторам об изменении их состояния – остатка. При этом сообщения могут генерироваться для акторов не обязательно в режиме онлайн, а в процессе чтения сообщений из RabbitMQ, как вариант, или базы данных. При этом каждое сообщение может параллельно в транзакции изменять некий показатель актора-отчёта. То есть таким образом отчётные показатели всё время пересчитываются. Это позволяет нам, помимо всего прочего, распределить вычисление итогового баланса во времени, то есть не в конце периода считаем всё и сразу и, как следствие, ждём несколько часов пока всё это закончится, а считаем баланс в процессе ведения учёта, непрерывно.

Теперь поговорим об удалённых от центрального офиса или базы данных узлах учёта. Давайте себе представим, что в центральном офисе установлена обычная 1С:Бухгалтерия с общим планом счетов.

В терминах сетевой игры план счетов – это набор персонажей. Удалённый узел учёта – это игрок. Предположим, что этот игрок хочет войти в игру с определённым набором персонажей. Он запрашивает у сервера этот набор или тот, который ему разрешено по статусу. Центральный сервер (уже в терминах 1С) отгружает ему метаданные соответствующих статей учёта, а также документы, которые могут с ними работать – только часть конфигурации. Далее узел учёта работает, выполняется обмен данными, отправляются сообщения различным акторам и так далее.

Дополнительно центральный сервер может отгружать определённые для данного узла учёта правила и бюджеты. Согласование новых объектов конфигурации или бюджетных лимитов может осуществляться при помощи Samunda BPMS или всё тех же акторов, запрограммированных как сервисы согласования.

Централизованная система метаданных и настроек полностью решает в такой модели проблему мастер-данных. Обмен данными и сообщениями может быть реализован как гибридная система, состоящая из традиционных таблиц СУБД и очередей сообщений, например, RabbitMQ или записей событий Kafka. Платформа Orleans используется для оперативного изменения и пересчёта балансовых, оперативных и прочих отчётных показателей.

В чём суть новизны системы.

Во-первых, это распределённая конфигурация или план счетов.

Во-вторых, это способ обмена данными при помощи Orleans.

В-третьих, это онлайн пересчёт отчётных (балансовых) показателей.