того, с ростом номера эпохи обучения величина прироста скорости обучения, участвующая в формуле (3.8), становится меньше и в итоге скорость обучения ограничена сверху двойным значением начальной скорости обучения.

Такое поведение сети говорит о наличии квазиустойчивых состояний, которые свойственны самоорганизующимся картам с топологическими дефектами. В рассматриваемой сети данный дефект проявляется в том, что каждый раз выключается из расчета количество нейронов, кратное числу единиц во входном сигнале с коэффициентом кратности $e, \mathrm{e} = 1,2,..m$.

С практической точки зрения это означает, что разработанная сеть может применяться для прокладки графиков движения. Однако существует не один стабильный режим, а три (устойчивое классическое; не сошедшаяся с заданными условиями, но выдавшая одно из возможных решений; стабильное функционирование на фиксированном уровне ошибки), ответы сети в двух из которых могут быть использованы на практике.

Литература

1. Knudsen E.I., S. duLac and S.D. Esterly. Computational maps in the brain.//Annual Review of Neuroscience, 1987. Vol. 10, p.41-65.

2. Хайкин С. Нейронные сети: Полный курс: Пер. с англ. Н. Н. Куссуль и А. Ю. Шелестова, под ред. Н. Н. Куссуль. — М.: Вильямс, 2006. — 1104 с.

3. А.В.Игнатенков, А.М.Ольшанский. Применение искусственной нейронной сети для построения расписаний процессов на примере графика движения поездов.// Современные информационные технологии и ИТ-образование. Т.2 (№11). 2015. //М., изд-во ВМК МГУ, 2015, - 614 с., с.50-55.

И.В. Осипов, Е.В. Симонова

ИССЛЕДОВАНИЕ АРХИТЕКТУРНЫХ РЕШЕНИЙ ПРИ ПОСТРОЕНИИ СЕРВИСА ПЛАНИРОВАНИЯ

(Самарский национальный исследовательский университет
имени академика С.П. Королева)

Введение

Рост промышленного производства мотивирует увеличение размера корпоративных систем, которые состоят из множества модулей, в том числе, модулей для построения стратегических и оперативных планов работ [1]. Увеличение масштабов систем требует анализа струкутры компонентов системы и их взаимодействия. Для построения быстрых, надежных и безопасных корпоративных систем используются различные современные средства, такие как очереди сообщений, легковесные балансировщики нагрузки, NOSQL базы данных, средства сжатия и шифрования данных.

Краткое описание предметной области

При построении системы планирования определяются характеристики задач, ресурсов, ограничений, а также способы взаимодействия клиентов с системой планирования. В общем случае, система планирования, вне зависимости от реализации, позволяет строить расписание, выдавать результат и наблюдать за процессом планирования.

Задача построения расписания относится к одному из разделов дискретной математики – теории расписаний. В общем случае, задача является NP-полной, поэтому при больших объемах данных продолжительность вычислений будет стремительно расти. Следовательно, вычислительные мощности должны расходоваться равномерно и независимо от основной системы.

Согласно современным тенденциям, наиболее широко используются мобильные способы взаимодействия пользователя с системой. Например, потребность в корректировке производственного плана может возникать не раз в квартал, а непосредственно во время совещания, поэтому скорость доработки расписания до допустимого варианта является критичной.

Постановка задачи

Необходимо спроектировать архитектуру системы планирования и взаимодействующих с ней сервисов таким образом, чтобы сервис планирования обеспечивал построение расписания, хранение допустимых результатов и оповещение о готовности плана.

В крупных корпоративных приложениях выделяют задачи, относящиеся к категории «hot spots» или «горячих точек»:

1. Кэширование.
2. Взаимодействие.
3. Аутентификация и авторизация.
4. Композиция.
5. Параллельные вычисления и транзакции.
6. Управление конфигурацией.
7. Связанность и сцепление.
8. Доступ к данным.
9. Работа с исключениями.
10. Ведение логов и мониторинг.
11. Поток операций.
12. Взаимодействие с пользователем.
13. Проверка данных.

При реализации сервиса планирования выделим функции композиции, взаимодействия между сервисами, кэширования и доступа к данным, образующие базис архитектуры системы планирования.

Сервис планирования

Сервисы планирования могут иметь различные реализации, детали которых следует скрывать с помощью единого интерфейса доступа. Основные способы взаимодействия внешних систем с сервисом планирования:

1. Передача информации в терминах абстрактной модели.

2. Запуск системы планирования.
3. Остановка планировщика.
4. Проверка состояния системы.
5. Подписка на результаты системы планирования.

Возможность параллельного выполнения задачи планирования обеспечивается дополнительным уникальным идентификатором, который характеризует пару: пользователь, сессия планирования.

Рассмотрим четыре категории «горячих точек» при построении сервиса планирования как отдельные задачи.

Композиция

При решении задачи композиции необходимо спроектировать слабосвязанную структуру и при этом сохранить возможность взаимодействия между модулями. Модули могут быть запущены в рамках отдельных серверов приложений на одной вычислительной машине или на разных машинах. Чтобы сохранить доступность системы в экстренных ситуациях, необходимо выполнить дублирование физических серверов, на которых расположены сервисы планирования, а затем выполнить балансировку задач между ними.

Балансировщик нагрузки – это программа, прослушивающая порт, через который внешние клиенты подключаются к службам. Такой подход скрывает структуру внутренней сети и предотвращает атаки на ядро, сетевой стек или несвязанные сервисы, работающие на других портах. Если серверная часть балансировщика недоступна, может быть выполнена переадресация на резервную систему балансировки.

Альтернативный подход к балансированию нагрузки не требует специального программного или аппаратного обеспечения и называется Round Robin DNS. В отличие от использования выделенной подсистемы балансировки нагрузки, эта методика предоставляет клиентам информацию о существовании нескольких серверов. Такой подход фактически раскрывает структуру подсети и не может быть использован, так как планировщик является ядром системы, которое должно быть защищено и скрыто от клиента.

Необходимо, чтобы балансировщик правильно перенаправлял запросы в зависимости от пользователя и его сессии планирования. Балансировщик должен иметь долговременное хранилище – базу данных (БД), в которой будет храниться информация о том, у какого пользователя, на каком сервере, какая сессия планирования выполняется. Скорость взаимодействия с хранилищем – наиболее критический фактор. Доступ к БД должен быть исключительно у балансировщика, а количество репликаций БД может быть равно количеству резервных балансировщиков. Резервные балансировщики и репликации БД желательно размещать в одних и тех же географических локациях, за счет чего скорость доступа к данным будет максимальной, а при переключении на резервный балансировщик хранилище из той же локации получает статус «мастер» (принцип master-slave).

Взаимодействие

При решении задачи взаимодействия возникают следующие вопросы:

1. Как передавать информацию между сервисами?
2. Как совершать асинхронные операции?
3. Как передавать секретные данные?

Сервис планирования отделен от системы управления, поэтому необходимо определить, каким образом происходит обмен данными. На данный момент наиболее распространенным и простым способом передачи данных по сети является протокол HTTP (HyperText Transfer Protocol).

Для асинхронных операций информирования о построении расписания в данном протоколе используются «вебхуки» (webhooks). Данный метод основан на вызове определенного HTTP-запроса при наступлении события.

Для передачи секретных данных HTTP протокол имеет расширение HTTPS (HyperText Transfer Protocol Secure). Протокол позволяет шифровать трафик и избегать атак в виде прослушивания канала передачи данных и атак типа «человек посередине». Для шифрования передаваемых данных используются протоколы SSL и TLS. Однако в 2014 году сообщалось об уязвимости текущих версий протокола SSL, поэтому большую популярность получает TLS. Протоколы базируются на использовании двух ключей: публичного и приватного. С помощью публичного ключа клиент шифрует передаваемые данные и только с помощью приватного ключа сервер может их расшифровать.

Кэширование

Для временного хранения требуется решить задачу кэширования результатов работы планировщика. В общем случае, требуется хранить идентифицирующую информацию сеанса планирования и расписание. Использовать базу данных для кэша подобного рода не имеет смысла. Следует выстроить иерархию «сервер/пользователь/расписание», имена файлов будут соответствовать номеру версии, тогда файл будет исключительно хранилищем результатов. Такие данные не подвергаются изменениям, поэтому имеет смысл сжимать результирующий план для экономии расхода памяти на хранение версий и более быстрой передачи по сети. Время сжатия не является критическим фактором. Качественное сжатие однородных данных демонстрирует алгоритм GZIP.

Взаимодействие с таким кэшем следует проводить, используя реализации очередей сообщений (ActiveMQ, RabbitMQ). Если сервер хранения данных будет недоступен, сами данные будут находиться в очереди.

Доступ к данным

Из описанной выше структуры следует, что исходные данные для планирования поступают к сервису по сети, а хранятся только результаты. Формирование исходных данных происходит с помощью сервиса менеджмента планирования. Сервис отвечает за всю работу с данными и является посредником для работы с системой планирования. Клиенты не могут напрямую отправить в систему планирования исходные данные. Сервис менеджмента может работать с различными клиентами (mobile, web, desktop) или серверами-посредниками средствами HTTPS протокола.

Заключение

Были исследованы архитектурные решения для построения сервиса планирования. На основании анализа выбраны способы балансирования нагрузки, передачи, сжатия и шифрования информации.

Литература

1. Е.В. Симонова, И.В. Осипов. Расширение модуля стратегического планирования цеха промышленного предприятия возможностями адаптивности // Труды международной научно-технической конференции «Перспективные информационные технологии (ПИТ-2016)», Самара, 26-28 апреля 2016 г. – Самара: Издательство Самарского научного центра РАН, 2016. – С. 311-315.

С.В. Платонов, А.В. Иващенко

ТЕСТИРОВАНИЕ И АНАЛИЗ КАЧЕСТВА МУЛЬТИАГЕНТНЫХ СИСТЕМ

(Самарский национальный исследовательский университет
имени академика С.П. Королёва г. Самара)

В настоящее время появляется всё больше и больше приложений с интегрированной в них мультиагентной системой, так же появляется необходимость в их тестировании.

Тестирование данных систем по стратегии чёрного ящика, в данном случае, малоэффективно из-за малой информативности данного подхода. При тестировании мультиагентных систем необходимо знать их концепцию.

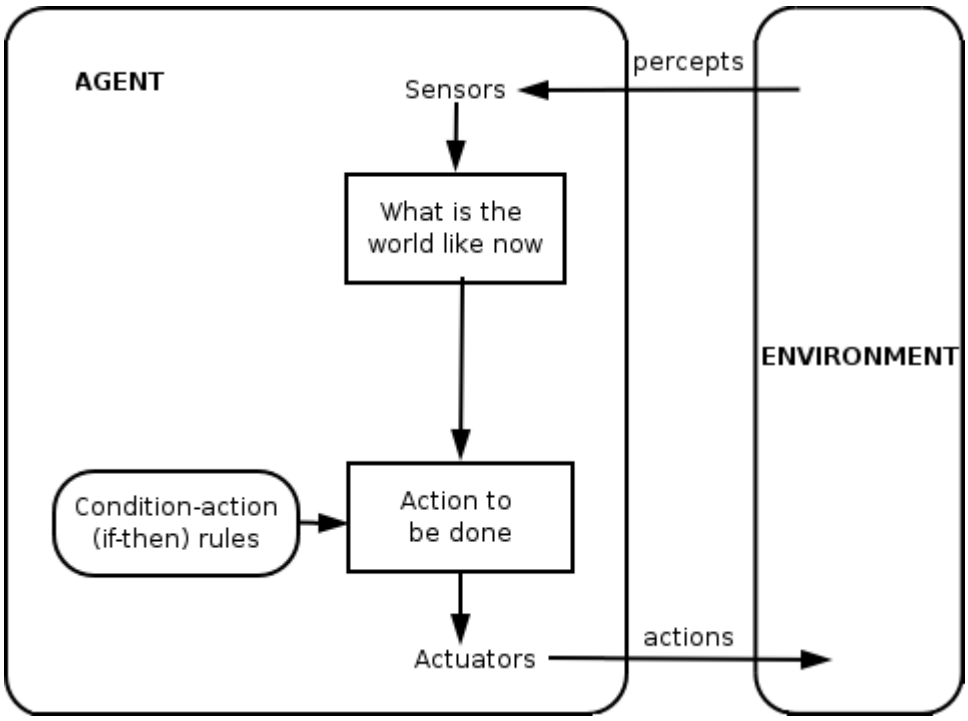

Рис. 1 – Схема мультиагентной системы