Тема: Конфигуратор мультимодульных программных систем

**Введение**

В рамках выпускной квалификационной работы будет рассмотрена идея применения конфигурационного моделирования для управления программной системой, включающей в себя множество модулей.

Рассмотрим понятие «мультимодульная программная система».

Мультимодульная означает, что система включает в себя группу модулей. Модуль может представлять собой ограниченный программный компонент, выполняющий определенную функцию. Каждый модуль имеет две основные характеристики: конфигурация и состояние. Модуль реализуется посредством множества алгоритмов. Алгоритмы детерминированы и не могут быть изменены в процессе эксплуатации системы. Алгоритмы оперируют данными и могут содержать параметры. Совокупный набор параметров является конфигурацией модуля. Конфигурация не задаёт поведение работы модуля, она лишь управляет параметрами, которыми оперирует программная логика модуля. Состояние содержит динамические параметры модуля. Состояние может меняться в процессе жизни модуля, при этом конфигурация модуля может быть изменена только извне.

Представим крупную программную систему, реализующую ряд требований для достижения множества целей. Как правило, подобным система свойственны следующие признаки:

1. Регулярные и частые обновления
2. Отказоустойчивость
3. Существенное количество модулей
4. Высокая стоимость
5. Высокий уровень декомпозиции функционала
6. Адаптивность

Далее расскажи про каждый признак, и приведи к мысли о том, как твой конфигуратор помогает достижению каждого из них.

**Анализ архитектуры мультимодульного ПО**

В основе архитектурны мультимодульных программных систем лежит понятие модулей и их взаимодействие. Модуль – архитектурная единица системы. Модули взаимосвязаны. Выделим основные типы связи модулей:

1. Посредством программного кода
2. Посредством использования артефакта модуля
3. Посредством сетевого взаимодействия

Далее расскажи про каждый тип связи и покажи их на схеме

Количество модулей в архитектуре современных программных систем может достигать порядка 50-и модулей со своей уникальной конфигурацией. Подобным объёмом модулей практически невозможно управлять без применения дополнительных средств обслуживания и конфигурирования. Данная причина является обоснованием к необходимости проектирования и создания ряда обязательных для системы модулей. Выделим основные среди них:

1. Модуль управления конфигурациями
2. Модуль распределения сетевых запросов
3. Модуль обслуживания
4. Модуль мониторинга и сбора журналов событий
5. Модуль выполнения отложенных задач

Далее расскажи про каждый модуль и покажи их на схемах (можно под каждый свою, можно – на одной)

Адаптация мультимодульной программной системы

Тут будет то, что вы писали в статье (кстати, ее все же опубликовали, нов elibrary я ее не нашел)) потом могу скинуть сборник)

Загляни в книжку Растригина (<http://www.mtas.ru/Library/uploads/1189332881.pdf>). Там есть что подобрать сюда. Также пни Макса, чтобы скинул тебе что-нибудь про его «решатель». Он тож с Нечаем делал на эту тему статьи.

Конфигурационное моделирование программных систем

Тут напишу, что для адаптированной системы можно применить конфигуратор и тд (там один из пунктов в ваших статьях)

Описание механизма управления конфигурациями

Рассмотрим концептуальную модель конфигурируемой программной системы, состоящей из множества модулей.

где

1. M – множество модулей
2. SM – множество состояний модулей
3. CM – множество конфигураций модулей
4. DM – взаимосвязи модулей
5. B – балансировщик запросов между модулями
6. C – конфигуратор

Основная реализация механизма управления конфигурациями ложится на плечи конфигуратора и балансировщика и осуществляется посредством асинхронного обновления параметров на стороне конфигуратора с последующим сигналом на запрос обновленной конфигурации ко всем модулям. Схематично данный процесс можно отобразить следующим образом:

Здесь покажи схему последовательных операций с её описанием (<https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8>)

Анализ процессов управления конфигурациями

Рассмотрим систему, состоящую из определенного количества модулей: 5 и взаимосвязанных систем: 2, а затем распишем на её примере основные бизнес-процессы, в которых участвуют операции по управлению конфигураций.

Определим инфраструктурную модель системы:

Здесь схема кластера. Покажи кластер с несколькими профилями (blue, green, standalone).

Детализируем основной набор процессов обслуживания и конфигурирования, применимый к данной системе:

1. Обновление профиля
   1. Гашение модулей
   2. Обновление модулей
   3. Обновление конфигурации профиля
   4. Поднятие модулей
   5. Получение конфигурации
2. Обновление приложения (blue-green схема)
3. Обновление конфигурации модуля в рантайме
4. Деактивация сервиса и метода
5. Перевод интеграционных модулей в режим заглушки

И описываешь процессы

Функциональные требования программной системы

Какие требования конкретно конфигуратор должен выполнять, короче, что он конкретно должен уметь.

Распиши здесь списком требования. После по каждому требованию сценарий использования с указаниями ролей (где админ, где модуль). Отдельно покажи сценарий по изменению конфигурации активации сервисов, то есть сценарий, где мы деактивируем сервис/метод.

Разработка программного продукта «Конфигуратор мультимодульных программных систем»

Здесь общее описание того, что конфигуратор как кусок кода будет делать.

По сути схемка с человечком-админом, который загружает JSON, а тот уже летит в конфигуратор, в роксу, в модули и описание этой схемки.

Здесь уже можно оперировать понятиями JSON, GRPC, HTTP и так далее.

Проектирование и дизайн пользовательского интерфейса

Здесь основные формы, действия, переходы по интерфейсу и скрины

Разработка пользовательского интерфейса

Здесь расскажи про технологию, про запросы, примеры запросов и описание входных и выходных данных

Разработка серверной части

Инфраструктурная схемка. Модули, конфигуратор, бд-шенька, какие где протоколы и всё такое. Расскажи про принцип работы ADK протобаф модулей (клиент, сервак). Почему выбрали протобаф. Его принцип работы. Про сервисную модель, про Flow (GRPC -> Spec -> Service -> Proxy -> GRPC).

Проектирование базы данных (RocksDB)

Концептуальная схема сущностей, параметры их. Расскажи про Value, что он из себя представляет, какие имеет формы. И дополнительно про типы данных конфигурации. По сути основные возможности конфигурации как типа данных. Также добавь про плюшки роксы.

Обеспечение безопасности передачи данных и управления модулями

Здесь расскажи про HTTPS SSL взаимодействие с конфигуратором и GRPC SSL между конфигуратором и модулями

Функциональное тестирование программной системы

Список основных тестов-спек. В идеале запилить на котспеках или грувиспоках и сюда скрины.

Нагрузочное тестирование программной системы

Здесь расскажи про гатлинг (<https://gatling.io/>). Он умеет и в GRPC тоже. Ну типо какую дал нагрузку, какие результаты получил. Пишется быстро, в принципе. Если долго/сложно, то сделай проще – Postman. Пошли лям запросов, здесь покажи стату из графаны по тому, как там что изменилось.

Анализ производительности системы

Расскажи здесь про мониторинг. Про графану, прометея, заббикс (Zabbix). Про то, как с их помощью можно смотреть на нагрузку. Дополнительно напиши, какие мы знаем способы оптимизации нагрузки. И можешь попробовать составить какую-нить формулу прогноза нагрузки. То есть, сколько будет выдерживать конфигуратор при наличии таких-то модулей с частотой обращения такой-то. Расскажи про кеш модулей здесь.

Заключение

Список литературы