RFT_CS

Rocket fuel and trajectory computing system

Автор: Булат Насыров

30 декабря 2022 г.

Оглавление

Предисловие

О чем будет идти речь, когда будем обсуждать ПО?
Так же как любая метафора, описание программного обеспечения с точки зрения архитектуры может что-то скрыть, а что-то, наоборот, проявить; может обещать больше, чем давать, и давать больше, чем обещать.
Оснавная привлекательность архитектуры - это структура. А структура - это то, что доминирует над парадигмами и суждениями в мире разработки ПО - компонентами, классами, функциями, модулями, слоями и службами, микро или макро. Но макроструктура многих программных систем часто пренебрегает убеждениями или пониманием - организация советских предприятий, невероятные небоскребы-башни (манареты) Дженга, достигающие облаков, археологические слои, залегающие в горной породе. Структура ПО не всегда интуитивно очевидна, как структура зданий.
Здания имеют очевидную физическую структуру, независимо от материала, из которого они построены, от их высоты или ширины, от их назначения и от наличия или отсутствия архитектурных украшений. Их структура мало чем отличается - в значительной мере она обусловлена законом тяготения и физическими свойствами материалов. ПО, напротив, никак не связано с тяжестью, кроме чувства серьезности. И из чего же сделано ПО? В отличие от зданий, которые могут быть построены из кирпича, бетона, дерева, стали и стекла, программное обеспечение строится из меньших программных компонентов, и т.д., вплоть до основания. Говоря об архитектуре, можно сказать, что программное обеспечение по своей природе является фрактальным и рекурсивным, выгравированным и очерченным в коде. Здесь важные все детали. Переплетение уровней детализации также вносит свой вклад в архитектуру, но бессмысленно говорить о ПО в физических масштабах. Программное обеспечение имеет структуру - множество структур и множество их видов, - но их разнообразие затмевает диапазон физических структур, которые можно увидеть на примере зданий. Можно даже довольно убедительно утверждать, что при проектировании ПО архитектуре уделяется куда больше внимания, чем при проектировании зданий, - в этом смысле архитектура ПО является более многообразной, чем архитектура зданий!
Но физический масштаб привычнее людям, и они часто ищут его в окружающем мире. Несмотря на привлекательность и визуальную очевидность, прямоугольники на диаграммах PowerPoint не являются архитектурой ПО. Да, они представляют определенный взгляд на архитектуру,

значит не получить ни общей картины, ни понятия об архитектуре: архитектура ПО ни на что не похожа. Конкретный способ визуализации - не более чем частный выбор. Этот выбор основан на следующем наборе вариантов: что включить; что исключить; что подчеркнуть формой или цветом; что, наоборот, затенить. Никакой взгляд не имеет никаких преимуществ перед другим.
Возможно, нет смысла говорить о законах физики и физических масштабах применительно к архитектуре ПО, но мы действительно учитываем некоторые физические ограничения. Скорость работы процессора и пропускная способность сети могут вынести суровый приговор производительности. Объем памяти и дискового пространства может ограничить амбиции любого программного кода. ПО можно сравнить с такой материей, как мечты, но ему приходится работать в реальном, физическим мире.

> *В любви, дорогая, чудовищна только безграничность воли, безграничность воли, безграничность желаний, несмотря на то, что силы наши ограничены, а осуществление мечты - в тисках возможности.*

Вильям Шекспир

Физический мир – это мир, в котором мы живем, в котором находятся и действуют наши компании и экономика. Это дает нам другой подход к пониманию архитектуры программного обеспечения, позволяющий говорить и рассуждать не в терминах физических законов и понятий.

> *Архитектура отражает важные проектные решения по формированию системы, где важность определяется стоимостью изменений.*

Гради Буч

Время, деньги, трудозатраты – вот еще одна система координат, помогающая нам различать большое и малое и отделять относящееся к архитектуре от всего остального. Она также помогает дать качественную оценку архитектуре – хорошая она или нет: хорошая архитектура отвечает потребностям пользователей, разработчиков и владельцев не только сейчас, но и продолжит отвечать им в будущем.

> *Если вы думаете, что хорошая архитектура стоит дорого, попробуйте плохую архитектуру.*

Брайан Фут и Джозеф Йодер

Типичные изменения, происходящие в процессе разработки системы, не должны быть дорогостоящими, сложными в реализации; они должны укладываться в график развития проекта и в рамки дневных или недельных заданий.
Это ведет нас прямиком к большой физической проблеме: путешествиям во времени. Как узнать, какие типичные изменения будут происходить, чтобы на основе этого знания принять важные решения? Как уменьшить трудозатраты и стоимость разработки без машины времени и гадания на кофейной гуще?

> *Архитектура – это набор верных решений, которые хотелось бы принять на ранних этапах работы над проектом, но которые не более вероятны, чем другие.*

Ральф Джонсон

Анализ прошлого сложен; понимание настоящего в лучшем случае переменчиво; предсказание будущего нетривиально.
К цели ведет много путей.
На самом темном пути подстерегает мысль, что прочность и стабильность архитектуры зависят от строгости и жесткости. Если изменение оказывается слишком дорогостоящим, оно отвергается – его причины отменяются волевым решением. Архитектор имеет полную и безоговорочную власть, а архитектура превращается в антиутопию для разработчиков и постоянный источник недовольств.
От другого пути исходит сильный запах спекулятивной общности. Он полон догадок, бесчисленных параметров, могильников с «мертвым» кодом и на нем подкарауливает множество случайных сложностей, способных покачнуть бюджет, выделенный на обслуживание.
Но самый интересный – третий, чистый путь. Он учитывает природную гибкость программного обеспечения и стремится сохранить ее как основное свойство системы. Он учитывает, что мы оперируем неполными знаниями и, будучи людьми, неплохо приспособились к этому. Он играет больше на наших сильных сторонах, чем на слабостях. Мы создаем что-то и совершаем открытия. Мы задаем вопросы и проводим эксперименты. Хорошая архитектура основывается скорее на понимании движения к цели как непрерывного процесса исследований, а не на понимании самой цели как зафиксированного артефакта.

Архитектура – это гипотеза, которую требуется доказать реализацией и оценкой.

Том Гилб

Чтобы пойти по этому пути, нужно быть усердным и внимательным, нужно уметь думать и наблюдать, приобретать практические навыки и осваивать принципы. Сначала кажется, что это долгий путь, но в действительности все зависит от темпа вашей ходьбы.

Поспешай не торопясь.

Роберт С. Мартин

Получайте удовольствие от путешествия.

Глава 1

Введение

1.1 Концепция

RFT_CS (Rocket fuel and trajectory computing system) Система расчета ракетного топлива и траектории полета ракеты - это Python-библиотека для разработки математических моделей. RFT_CS изначально был спроектирован так, чтобы его можно было внедрять постепенно. Другими словами, **вы можете начать с малого и использовать только ту функциональность RFT_CS, которая необходима вам в данный момент**. Также в случае, если вам нужно изменить поведение/вычисления функции, есть возможность конфигурации методов под ваши нужды.

1.2 Цель

Основная цель - создать математическую модель процессов, связанных с полётом одно и многоступенчатых, твердо и жидко топливных ракет и для вычисления траектории полёта баллистических ракет. Данное ПО может быть использовано для создания космических/баллистических ракет или своих научных экспериментов.

1.3 Технологии

Программное обеспечение построено на высокоуровневом языке программирования Python и отдельные микропроцессоры написаны на языке C. Также для сложных математических вычислений использовались библиотеки, специально созданные для этой цели.

1.4 Декомпозиция задачи

Начнём с составных частей ракеты, а также внешние факторы, влияющие на полёт. Начнём с состава космической ракеты:

1.5 Обработка ошибок

Могут возникать ошибки связанных с некорректным математических операций, перегрузкой или долгим ожиданием ответа от системы и неверным вводом/выводом данных, и др. Подобные ошибки обрабатываются и выводятся в виде ответа, а также записываются в логи. "**При добавлении новых функций или использование встроенных функций важно не забывать обрабатывать ошибки и добавлять логирование для дальнейшего удобства исправления багов.**"

Глава 2

Установка

2.1 Создание среды Linux/MacOS

Для создания виртуального окружения вам понадобиться установленный Poetry..., Python версии 3.9. После установки из GitHub, вы открываете рабочую директорию RFT_CS. *poetry install*, после загрузки вы пишите... *source .venv/bin/activate* Υ!

2.2 Запуск программы Linux/MacOs

Чтобы запустить программу, вы должны зайти в директорию RFTCS $\Rightarrow$: *python3main.py*.

2.3 Создание среды Windows

Для создания виртуального окружения вам понадобиться установленный Poetry..., Python версии 3.9. После установки из GitHub, вы открываете рабочую директорию RFT_CS. poetry install, после загрузки вы пишите....venv\Scripts \activate Υ!

2.4 Запуск программы Windows

Чтобы запустить программу, вы должны зайти в директорию RFTCS написать: python main.py и уже заполнять данные в формате целого или дробного числа.

2.5 Создание среды с использованием Makefile

Для этого достаточно перейти в директорию .INSTALL/install_env и написать. . . make

Глава 3

Численное моделирование топлива ракеты

3.1 Описание

Численное моделирование топлива ракеты. Производяться расчеты связанные с кольчеством топлива на разных стадиях топлива.

3.2 Импорты

Импортируется библиотеки numpy и matplotlib, и typing. Из файла format импортируется функция main_rocket_format, для более удобного вывода результа в консоль или json файл.

3.3 Константы

- *Ускорение свободного падения - 9.81*
- *Коэффициент массы конструкции для единицы массы топлива - 400*

3.4 Функции

- *natural_logarithm (Функция нахождения натурального логарифма) - принимает 2 параметра (полную массу рокеты с топливом и без топлива, с типом float), выводит число типа float.*

- *euler (Функция расчет с помощью Эйлерова числа E) - принимает 2 параметра (Общую скорость и удельный импульс, с типом float), выводит число типа float.*
- *total_speed (Сумма всех скоростей) - принимает 3 параметра (Удельный импульс, полную массу рокеты с топливом и без топлива, с типом float), выводит число типа float.*
- *total_oil (Функция для расчета топлива) - принимает 3 параметра (Удельный импульс, общая скорость и массу ракеты без топлива, с типом float), выводит число типа float.*
- *massa_construction_rocket (Функция расчета массы конструкции ракеты) - принимает 1 параметр (полную массу рокеты с топливом, с типом float), выводит число типа float.*

Глава 4

Математическое моделирование движения тел

4.1 Описание

Математическое моделирование движения тел. Определяет траекторию полёта ракеты.

4.2 Импорты

- *Импортируется библиотеки numpy и typing.*
- *Из файла rocket_flight_simulation импортируется функция resistance_force_env, для расчета лобового сопротивления.*
- *Из файла rocket_fuel_calculation импортируется функция total_oil, для получения общего количества топлива в ракете.*

4.3 Константы

- *Расход топлива*
  - *1 т/с*
- *Скорость истечения газов*
  - *2/3 км/с*
- *Ускорение свободного падения*

– *9.81*

- *Коэффициент лобового сопротивления*
  - 0.14

4.4 Функции

- *calculation_rocket_movement (Математическая модель движения тел с переменной массой, для вертикального взлета) - принимает 2 параметра (скорость ракеты и сопротивления, с типом float), выводит число типа float.*

- *vector_speed (Горизонтальная составная скорость) - принимает 2 параметра (радиус и высоту, с типом float), выводит число типа float.*

- *rocket_flight_description (Модель для описания полета ракеты) - принимает 5 параметров (тета и скорость ракеты, общую массу, сопротивления и вектор скорости, всё с типом float), выводит числа типа float, в виде списка [скорость ракеты, основную тета, y, x].*

- *rocket_mass (масса ракеты) - это условие, которое проверяет меньше ли настоящая масса (из - за расхода топлива), чем масса конструкции. Если меньше, то от начальной массы отнимают произведение расхода топлива и времени. Иначе масса конструкции = настоящая масса ракеты.*

Глава 5

Численное моделирование полета ракеты

5.1 Описание

Численное моделирование полета ракеты. Производяться расчеты связанные с полётом ракеты в земной среде. Используются физические явления: поступательное и вращательное движение; гравитация; зависимость плотности воздуха от высоты; реактивная сила с отклонением вектором тяги; сопротивления воздуха.

5.2 Импорты

- *Из файла rocket_fuel_calculation.py импортируется функция total_speed, для получения дельта скорости (общей скорости).*
- *Из файла format импортируется класс FlightFormat, для более удобного вывода результа в консоль.*

5.3 Константы

- *Плотность воздуха, кг/м*^{3} $-$ 1.27, / $-$ 0.029
- *Температура, K - 300*
- *Ускорение свободного падения - 9.81*
- *Ро - плотность - 16900*

- *Коэфициент формы - 1.2*
- *Коэффициент лобового сопротивления - 0.14*

5.4 Функции

- *distance_N_step (Функция расстояния от горящей поверхности топлива до стенки камеры сгорания) - принимает 2 параметра (расход топлива и шаг, с типом float), выводи число типа float.*
- *tg_Beta (Функция угол полета ракеты относительно поверности земли) - принимает 1 параметр и 2 константы (скорость, радиус земли и начальный радиус старта ракеты, с типом float), выводит число типа float.*
- *elliptical_range (Функция эллиптической дальности полета) - принимает 1 параметр и 1 константу (скорость и радиус Земли, с типом float), выводит число типа float.*
- *mass_rocket (Функция массы ракеты) - принимает 2 параметра (масса пустой ракеты и топлива ракеты, с типом float), выводит число типа float.*
- *amount_gas_released (Функция количества выделяемого газа за 1 моль) - принимает 1 параметр и 1 константу (масса и средняя молярная масса, с типом float), выводит число типа float.*
- *overpressure (Функция избыточного давления в камере сгорания на n-шаге) - принимает 1 параметр и 3 константы (расход, средняя молярная масса, универсальная газовая константа и температура горения, с типом float), выводит число типа float.*
- *thrust_force (Функция силы тяги) - принимает 1 параметр и 1 константу (давление и площадь поперечного сечения, с типом float), выводит число типа float.*
- *impuls (Функция импульса, сообщаемого ракете на n-шаге), принимает 2 параметра (давление и время, с типом float), выводит число типа float.*
- *height_force (Функция высоты ракеты над стартовой площадкой) - принимает 3 параметра (высота площадки, высота ступени, количество ступеней, с типом float), выводит число типа float.*

5.5 Классы

- *Resistance (Класс для расчета сопротивления) - принимает 3 параметра (скорость, сила тяги и масса, с типом float). Включает функции:*
  - *_aerodynamic_pressure (Скрытая функция аэродинамического напора) - принимает 1 параметр и 1 константу (скорость и плотность воздуха, с типом float), выводит число с типом float.*
  - *aerodynamic_drag (Функция аэродинамического сопротивления) - принимает 1 параметр и 1 константу (аэродинамическое сопротивление и площадь поперечного сечения, с типом float), выводит число типа float.*
  - *gravitation_losses (Функция гравитационные потери) - принимает 2 константы (ускорение свободного падения и угол вектора тяги над горизонтом, с типом float), выводит число типа float.*
  - *control_losses (Функция потери скорости на управление) - принимает 2 параметра и 1 константу (сила тяги, массу и угол между векторами тяги, и скорости ракеты, с типом float), выводит число типа float.*
- *Speed (Класс для расчета скорости) - принимает 5 параметров (силу тяги, гравитционные потери, масса, время, начальная скорость, с типом float). Включает функции:*
  - *_resultant_force (Скрытая функция равнодействующая сила) - принимает 3 параметра и 1 константу (силу тяги, гравитационное сопротивления, массу и ускорение свободного падения, с типом float), выводит число типа float.*
  - *rocket_acceleration (Функция ускорение ракеты) - принимает 2 параметр (равнодействующая сила и масса, с типом float), выводит число типа float.*
  - *rocket_speed (Функция скорости ракеты) - принимает 3 параметра (начальная скорость, ускорение ракеты и время, с типом float), выводит число типа float.*
- *ModelFlight (Класс для моделирования полета) - принимает 4 параметра (масса, начальная скорость, время и сила тяги, с типом float). Включает функции:*

– *_total_resistance (Скрытая функция сумма сопротивлений), принимает 4 параметра (скорость, сила тяги, масса, с типом float), выводит число типа float.*

– *_total_speed (Скрытая функция сумма скоростей), принимает 5 параметров (сила тяги, гравитационное сопротивление, масса, время, начальная скорость, с типом float), выводит типа float.*

– *_total_distance (Скрытая функция общая дистанция), принимает 1 параметр (общая скорость, с типом float), выводит с типа float.*

– *model_stack (Функция списка параметров), принимает 3 параметра (сумма сопротивлений, сумма скоростей и общая дистанция, с типом float), выводит 4 значения (сопротивление, скорость, расстояние, угол полета ракеты) с типами float.*