МИНОБРНАУКИ РОССИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

09.03.01 Информатика и вычислительная техника.
Программная инженерия и компьютерные науки

Разработка прототипа облачной IDE для системы LuNA

Выполнил: Александров Евгений Александрович
студент ФИТ НГУ
Руководитель ВКР: Малышкин Виктор Эммануилович
зав. каф. ПВ ФИТ д.т.н., профессор
Соруководитель ВКР: Перепёлкин Владислав Александрович
ст. преп. каф. ПВ ФИТ НГУ
Рецензент: Черкасов Александр Владимирович
руководитель проектов, ООО «Скайтек»

Новосибирск, 2021

*Актуальность проблемы

Проблемы установки необходимого ПО, хранения файлов, обновления софта актуальны при разработке программ в наше время. Облачная IDE решает эти проблемы.
Создание облачной IDE - это комплексная проблема, включающая в себя разработку архитектуры облачной IDE, требований, которые удовлетворяют потребностям пользователей, а также удобного, интуитивно понятного для пользователя интерфейса.

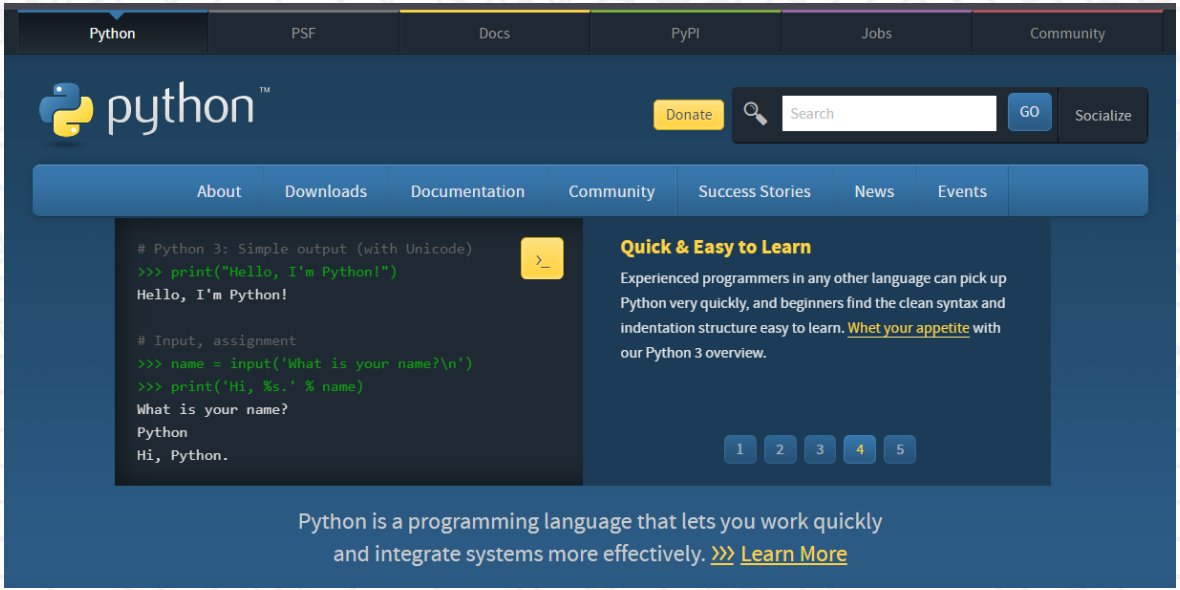
*LuNA

LuNA (Language for Numerical Algorithms) - система фрагментированного программирования, предназначенная для поддержки параллельной реализации больших численных моделей для суперкомпьютеров. LuNA нуждается в такой облачной IDE.

В **группу пользователей** суперкомпьютеров входят специалисты в своих предметных областях, которые используют численное моделирование для решения поставленных перед ними задач.

Для специалистов нужен продвинутый функционал: запуск на кластерах, использование системы очередей, хранение файлов, отладчик, взаимодействие с системой визуализации фрагментированной программы.

*Обзор существующих IDE (1 из 2)



python.org – официальный сайт, где можно познакомиться с Python. Сразу бросается в глаза самый большой элемент на сайте – командная строка с пятью примерами, с помощью которых, можно познакомиться с синтаксисом, операциями, работой со списками, определением функций.

*Обзор существующих IDE (2 из 2)

play.golang.org – сайт, где можно познакомиться с возможностями этого языка и запустить предлагаемые примеры. Интерфейс гораздо проще чем у python.org, снизу текстовой поле, куда нужно вписывать код, а на панели сверху – несколько управляющих кнопок.

*Выводы по существующим IDE

1. Наличие поля ввода исходного кода
2. Возможность просмотра тестовых примеров
3. Подсветка синтаксиса
4. Возможность перехода к документации
5. Запуск программы и возможность видеть результат её исполнения

*Цели и задачи

Цель работы - разработать облачную IDE для системы фрагментированного программирования LuNA.

Задачи

1.Разработать требования к IDE.
2.Разработать функциональность.
3.Разработать архитектуру IDE.
4.Реализовать front-end часть IDE.
5.Разработать вычислительный модуль.
6.Провести тестирование.

*Устройство работы LuNA-программы

На вход программе подается набор входных файлов одного из двух типов: **файл с расширением ".fa"** (может быть несколько), в нем описывается LuNA программа, **и файл с расширением ".cpp"**, который реализует последовательные процедуры, осуществляющие вычисление выходных из входных данных.

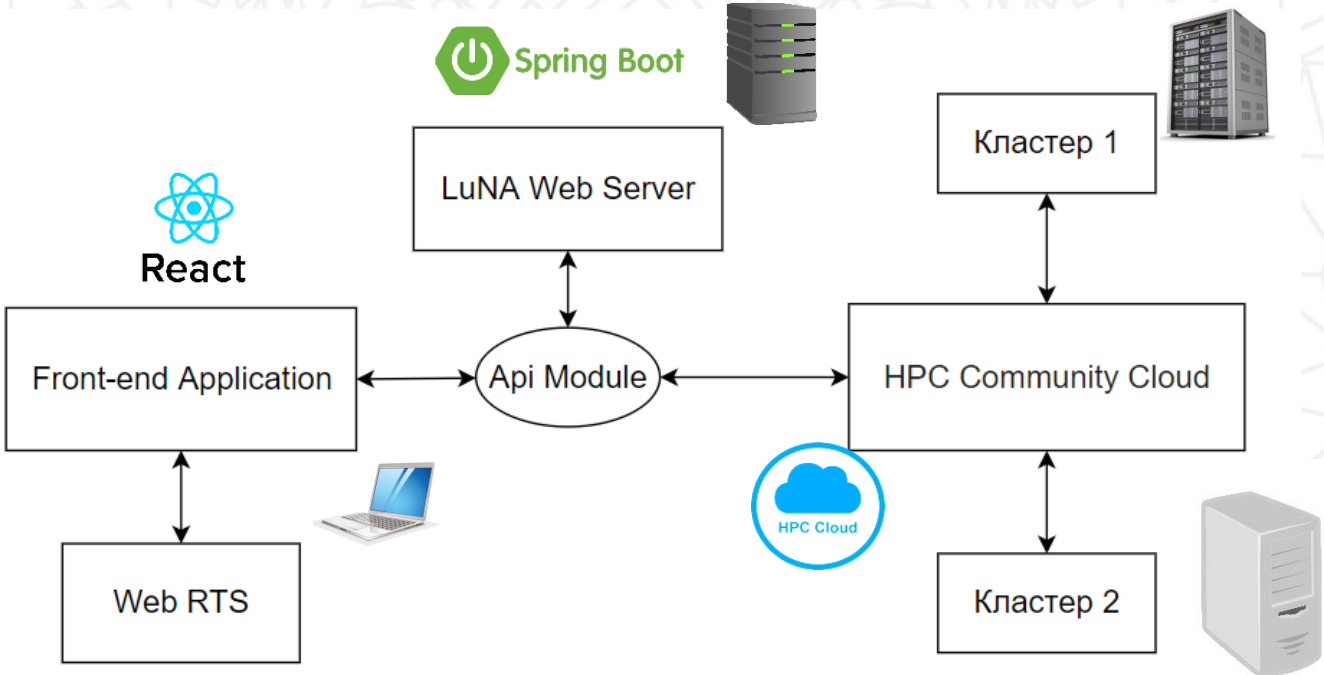
Результат компиляции – это файл libucodes.so - исполняемое представление, скомпилированное обычным компилятором.

*Требования к облачной IDE

Специфичные требования:

1. Запуск на различных вычислительных системах.
2. Требования к количеству одновременных подключений к IDE.
3. Просмотр и запуск тестовых примеров.
4. Интеграция существующего инструментария
5. Отображение результатов производительности.
6. Возможность расширения.

* Архитектура решения



Реализация Front-end Application

LuNA Cloud IDE

localhost:8000

LuNA Cloud IDE

LuNA Home Documentation Examples LuNA Web Server Run

test.fa

```
/*
 Initialization of data fragments and data dependencies.
*/

import c_init(int, name) as init;
import c_iprint(int) as iprint;

sub main() {
    df x;

    cf a: init(7, x) @ {
        //      req_count x=1;
        locator_cyclic: 1;
    };

    iprint(x) @ {
        request x;
        locator_cyclic: 2;
    };

}
@ {
    locator_cyclic x => 3;
}
```

ucodes.cpp

```
#include <cstdio>

#include "ucenv.h"

extern "C" {

    void c_init(int val, OutputDF & df) {
        df.setValue(val);
        printf("c_init: %d, size: %d\n", val, (int) df.getSize());
    }

    void c_iprint(int val) {
        printf("c_iprint %d\n", val);
    }

    void c_print(const InputDF & df) {
        printf("c_print: %d\n", df.getValue < int > ());
    }

}
```

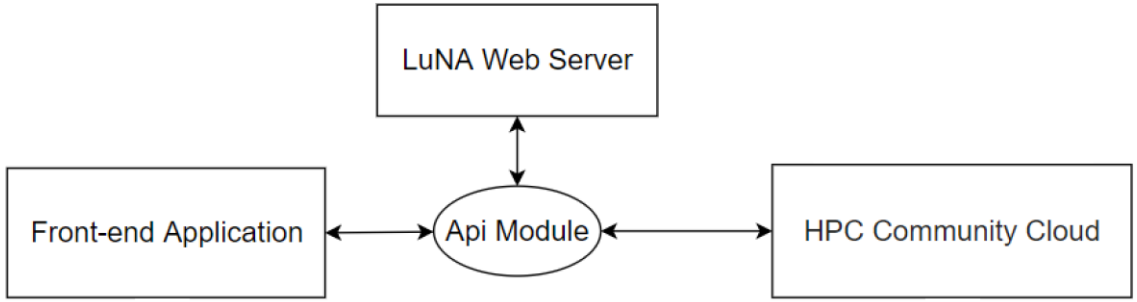
Output

Table

Clear

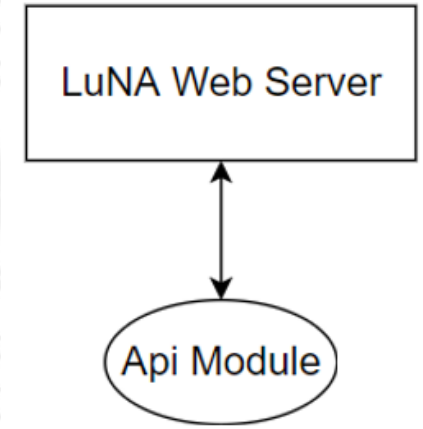
*Api Module

Api Module - модуль через который облачная IDE взаимодействует с вычислительными системами, например, LuNA Web Server. Api Module является точкой расширения, для интеграции других вычислительных систем.



*Реализация LuNA Web Server

LuNA Web Server – вычислительная система с предустановленной LuNA. Реализована на Java с использованием фреймворка Spring. Spring - это фреймворк состоящий из множества модулей для различных задач. Мы используем Spring Boot – подсистема для построения веб-приложения.



LuNA Cloud IDE

localhost:8000

LuNA Cloud IDE

LuNA Home Documentation Examples LuNA Web Server Run

test.fa

```
/*
 Initialization of data fragments and data dependencies.
*/

import c_init(int, name) as init;
import c_iprint(int) as iprint;

sub main() {
    df x;

    cf a: init(7, x) @ {
        //      req_count x=1;
        locator_cyclic: 1;
    };

    iprint(x) @ {
        request x;
        locator_cyclic: 2;
    };

}
@ {
    locator_cyclic x => 3;
}
```

ucodes.cpp

```
#include <cstdio>

#include "ucenv.h"

extern "C" {

    void c_init(int val, OutputDF & df) {
        df.setValue(val);
        printf("c_init: %d, size: %d\n", val, (int) df.getSize());
    }

    void c_iprint(int val) {
        printf("c_iprint %d\n", val);
    }

    void c_print(const InputDF & df) {
        printf("c_print: %d\n", df.getValue < int > ());
    }

}
```

Output

```
luna: creating build dir
luna: preprocessing
luna: parsing
```

Table

Clear

* Анализ предложенного решения

Облачная IDE для системы LuNA гибкая в выборе вычислительной системы, за счет компонента архитектуры Api Module, что делает IDE слабосвязанной от конкретной вычислительной системы и является сильной стороной.

Дизайн облачной IDE в дальнейшем требует доработки, добавления анимации во время исполнения действий.

* Заключение

В результате была разработана архитектура и реализован прототип облачной IDE для системы фрагментированного программирования LuNA.

На защиту выносятся:

1. Требования к облачной IDE для системы LuNA
2. Архитектура облачной IDE
3. Реализация модулей Front-end Application, Api Module, LuNA Web Server

Дальнейшие планы:

1. Интеграция с HPC Community Cloud
2. Реализация задания аргументов программы
3. Интеграция с Web RTS

* Апробация работы

Данная работа была опубликована на 59-й Международной научно-студенческой конференции, г. Новосибирск, 2021 г.
Публикация по теме:
Александров Е. А. Разработка прототипа облачной IDE для системы фрагментированного программирования LuNA / Е. А. Александров // Информационные технологии : Материалы 59-й Междунар. науч. студ. конф. 12–23 апреля 2021 г. / Новосиб. гос. ун-т. – Новосибирск : ИПЦ НГУ, 2021. – С. 110.