

# Реализация графика отображения результатов LuNA программы

Исполнитель: Александров Евгений студент 4 курса ФИТ НГУ  
Руководитель: Перепёлкин Владислав Александрович

# Постановка задачи

Тема моей ВКР это разработка прототипа облачной IDE для системы LuNA, и в рамках этой зимней школы я ставлю следующие задачи:

1. Реализовать отображение графика результатов времени исполнения программ
2. Реализовать настройку количества потоков исполнения

## Идея решения

Для первой задачи я выбрал js библиотеку D3 визуализации данных, потому что она проста в интеграции и использовании для изменяющихся данных.

Для выбора количества потоков я придумал реализовать модальное окно, в котором задается этот параметр.

# Реализация

LuNA Home Documentation Examples ▾

LuNA Web Server ▾

Run

test.fa

```
1  #!/usr/bin/luna
2  /*
3   Matrix multiplication.
4   */
5
6  import c_init(int, name) as init;
7  import c_init_submatrix(int `file, int `row, int `col, int `height, int `width, name `de:
8  import c_mult_matrix(value `a, value `b, name `c) as mult_mat;
9  import c_sum_matrix(value `a, value `b, name `c) as sum_mat;
10 import c_save_submatrix(value `a, int `row, int `col) as save_mat;
11 import c_copy_matrix(value `a, name `b) as copy_mat;
12
13 C++ sub empty() ${<END}
14 {
15     // do nothing
16 }END
17
18 #define FG_SIZE 4
19 #define FG_COUNT 10
20
21 sub calc_mat(name A, name B, name C, int i, int j, int N)
22 {
23     df Ctmp, Csum;
24
25
```

ucodes.cpp

```
1  #include <cstdio>
2  #include <cstring>
3  #include "ucenv.h"
4
5  using namespace luna::ucenv;
6
7  struct Matrix
8  {
9      size_t Height, Width;
10     double *Data;
11 };
12
13 void create_matr_buf(OutputDF &out, const int &height, const int &width)
14 {
15     out.create(2*sizeof(size_t) + height*width*sizeof(double));
16
17     size_t *data = static_cast<size_t*>(out.get_data());
18     data[0]=height;
19     data[1]=width;
20 }
21
22 void *shift(void *buf, int ofs)
23 {
24     return (void*)((char*)buf)+ofs;
25
```

Output

```
1 luna: creating build dir
2 luna: preprocessing
3 luna: parsing
```

Table

1.0		
0.9	0.83186	0.860957

Clear

# Реализация

The screenshot shows the LuNA web interface. At the top, there is a navigation bar with 'LuNA', 'Home', 'Documentation', and 'Examples'. On the right, it says 'LuNA Web Server' and has a 'Run' button. The main area is a code editor displaying C++ code for matrix multiplication. A 'Variables' dialog box is open in the center, with the title 'Variables' and a field for 'Number of threads' set to '4'. Below the field are 'Close' and 'Run' buttons. The code in the background includes comments and function definitions for matrix operations.

```
test.fa
1  #!/usr/bin/luna
2  /*
3  Matrix multiplication.
4  */
5
6  import c_init(int, name) as init;
7  import c_init_submatrix(int `file, int `row, int `col, int `height, int `width) as sub;
8  import c_mult_matrix(value `a, value `b, name `c) as mult;
9  import c_sum_matrix(value `a, value `b, name `c) as sum;
10 import c_save_submatrix(value `a, int `row, int `col) as save_mat;
11 import c_copy_matrix(value `a, name `b) as copy_mat;
12
13 C++ sub empty() ${<END}
14 {
15     // do nothing
16 }END
17
18 #define FG_SIZE 4
19 #define FG_COUNT 10
20
21 sub calc_mat(name A, name B, name C, int i, int j, int N)
22 {
23     df Ctmp, Csum;
24
25     double *Data;
```

# Реализация

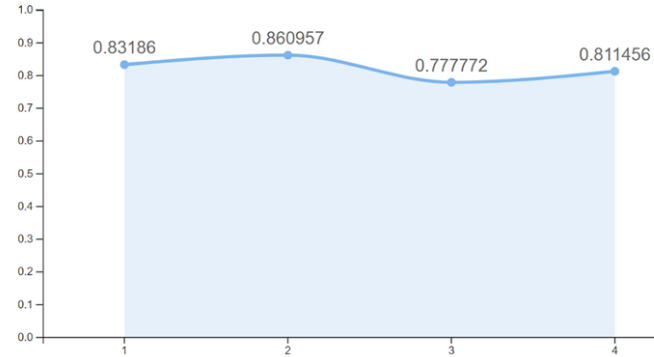
```
35 for k=2..N-1
36     cf sum[k]: sum_mat(Ctmp[k], Csum[k-1], Csum[k]);
37
38     copy_mat(Csum[N-1], C);
39 }
40
41 sub main()
42 f
```

```
18 data[0]=height;
19 data[1]=width;
20 }
21
22 void *shift(void *buf, int ofs)
23 {
24     return (void*)((char*)buf)+ofs;
25 }
```

## Output

```
1 luna: creating build dir
2 luna: preprocessing
3 luna: parsing
4 luna: let substitution
5 luna: generating cpp blocks
6 luna: generating recommendations
7 luna: generating cpp code
8 luna: constructing makefile
9 luna: building program
10 luna: running program
11 c_init: 4 --> (df), size: 4
12 c_init: 10 --> (df), size: 4
13 c_mult_matrix (df) (df) (df)
14 A: 4x4
15 B: 4x4
16 c_mult_matrix (df) (df) (df)
17 A: 4x4
18 B: 4x4
19 c_mult_matrix (df) (df) (df)
20 A: 4x4
21 B: 4x4
22 c_mult_matrix (df) (df) (df)
23 A: 4x4
24 B: 4x4
25 c_mult_matrix (df) (df) (df)
```

## Table

[Clear](#)

# Заключение

В рамках Зимней школы я выполнил поставленные задачи, а именно реализовал отображение график, отображающий время исполнения программ, и настройку количества потоков.

Дальнейшие планы:

1. Интеграция WebRTS
2. Интеграция HPC Community Cloud
3. Интеграция Визуализатора