Зимняя школа по параллельному программированию 2017

Распределенная система исполнения фрагментированных программ на базе Web-технологий

Выполнил: Ажбаков Артем, ФИТ НГУ

Руководитель: Перепелкин В.А.

03.02.2017

Введение: LuNA

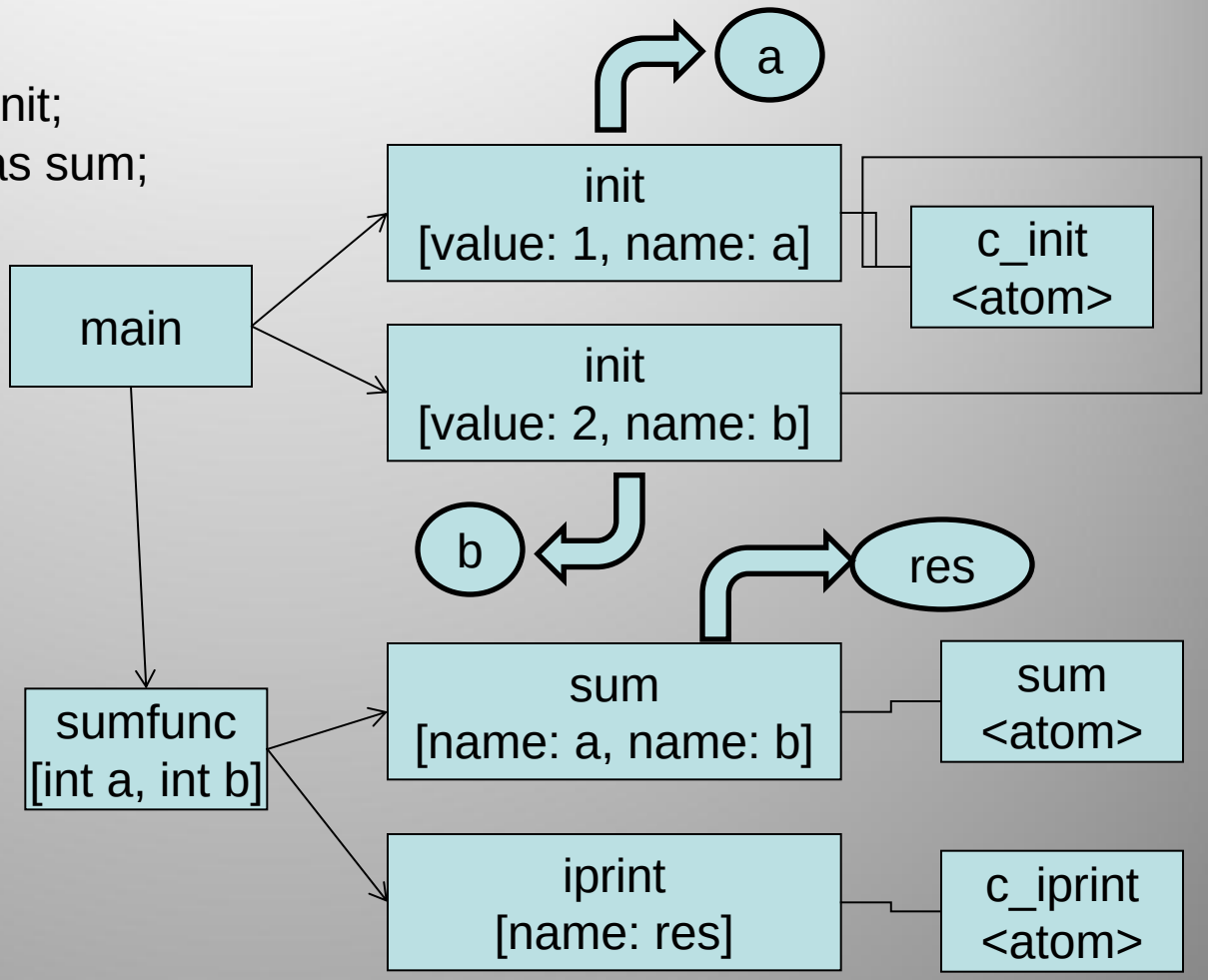
- LuNA – система автоматизации параллельного программирования
- Язык, компилятор, система исполнения
- Фрагментированное программирование

Введение : программа LuNA

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

sub sumfunc (int a, int b) {
    df res;
    sum (a, b, res);
    iprint(res);
}

sub main()
{
    df a, b, res;
    init(1, a);
    init(2, b);
    sumfunc(a, b);
}
```



Внутреннее представление программы

```
"iprint": {
   "type": "extern",
   "code": "c_iprint",
   "args":{
          "type": "int"
   }
}
```

```
"init": {
   "type": "extern",
   "code": "c_init",
   "args": {
     "type": "int",
     "type": "name"
   }
}
```

```
"sum": {
      "type": "extern",
      "code": "sum",
      "args": {
         {"type": "int"},
         {"type": "int"},
         {"type":
    "name"}
       }
   }
```

```
"main": {
 "body":
  {
    "code": "init",
    "args": [
     {"type": "iconst", "value": 1},
     {"ref": ["a"], "type": "id"}
     ],
    "type": "exec",
  },
  {
     …..
  }
}
```

Проблема

- Есть нереализованная возможность для улучшения системы за счет повышения переносимости.

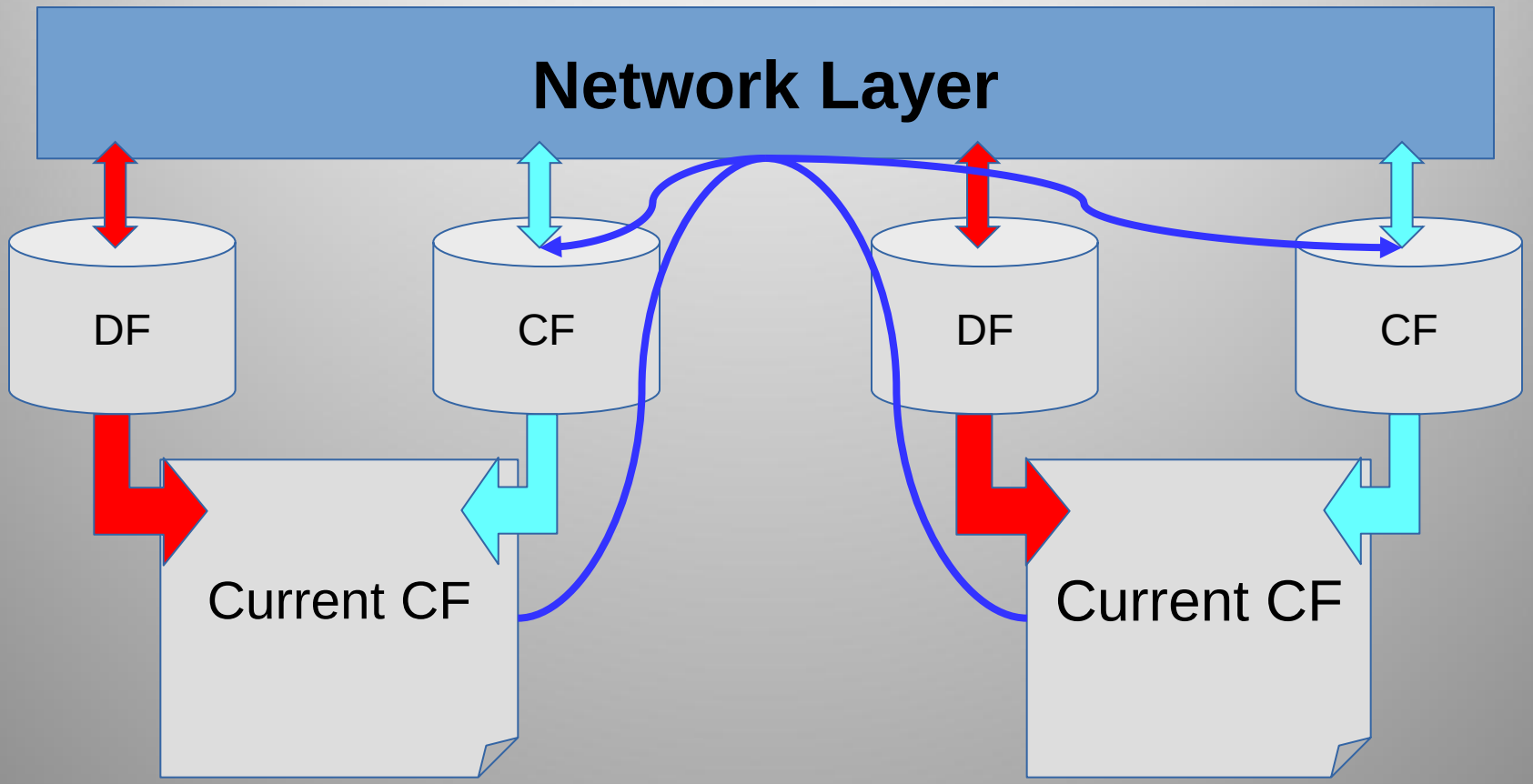
Решение

- Создание переносимой распределенной децентрализованной среды исполнения.

Было сделано:

- Написана программа на Javascript, исполняющая код во внутреннем JSON-представлении с использованием предоставленных атомарных функций.
- Разработан прототип системы сетевого peer-to-peer взаимодействия (WebRTC + nodejs) для двух исполняющих узлов.

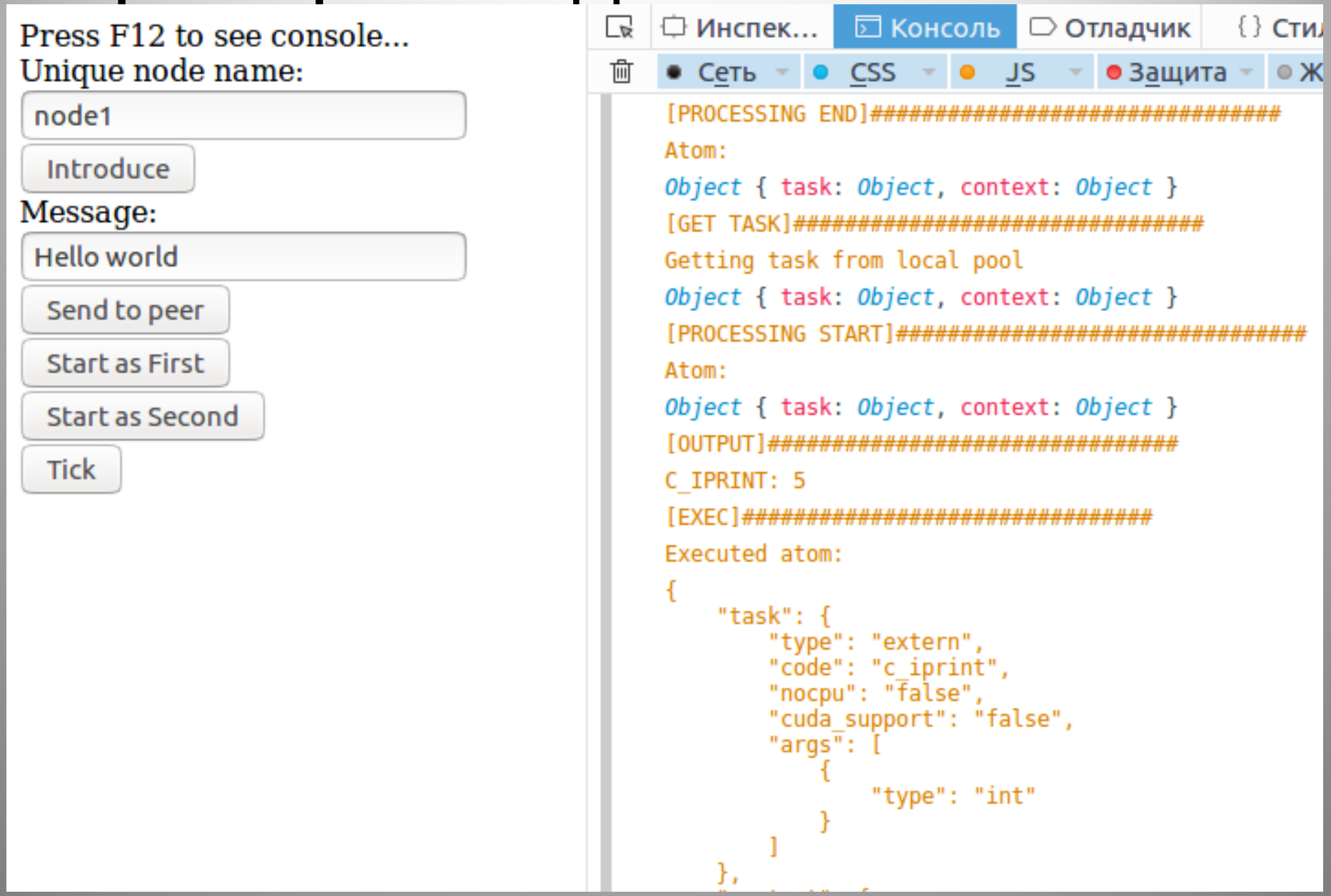
Было сделано:



Тестовая программа

```
1  /*
2   Subroutines and passing of arguments.
3  */
4
5  import c_init(int, name) as init;
6  import c_iprint(int) as iprint;
7
8  sub initialize(name x, int val)
9  {
10     init(val, x[6]);
11 }
12
13 sub display(name y, int idx)
14     iprint(y[idx*2][6]);
15
16 sub main()
17 {
18     df z;
19     initialize(z[10], 5);
20     display(z, z[10][6]);
21 }
22
```

Пример вывода



Результат

- Корректное исполнение программ во внутреннем представлении в одиночном и парном режимах
- Установка peer-to-peer соединения между исполняющими узлами
- Сигнальный сервер
- Текущая топология — пара узлов
- Журналирование

Заключение

- Таким образом, можно считать, что за время зимней школы был сделан шаг в разработке переносимой распределенной системы исполнения.
- Дальнейшее развитие подразумевает доработку прототипа исполняющего узла, реализацию новых топологий, обеспечение отказоустойчивости.

Спасибо за внимание

Ажбаков Артем, ФИТ НГУ