

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

# **Разработка и реализация переносимых алгоритмов распределённого исполнения фрагментированных программ**

Выполнил:

**Ажбаков Артем  
Альбертович**

Научный руководитель:

**Маркова Валентина Петровна**  
к.т.н., доцент кафедры ПВ ФИТ НГУ  
Старший научный сотрудник  
ИВМиМГ СО РАН

# Проблема интеграции разнородных вычислительных ресурсов

Современные вычислительные сети состоят из разнородных устройств - разработка параллельной программы усложняется, необходима автоматизация.

При выполнении параллельной программы в неоднородной среде особую сложность обретают задачи:

- Динамическая балансировка
- Эффективная передача данных
- Сборка мусора
- Отказоустойчивость
- Другие

**Необходима программная платформа для экспериментальных исследований.**

# Обзор: существующие системы автоматизации параллельного программирования

Для экспериментальной платформы важно:

- Переносимость
- Открытость программной архитектуры
- Поддержка использования динамических и статических свойств программы

Charm++, OpenTS, GRID Superscalar:

- + применяется фрагментирование, поддерживаются неоднородные вычислители, осуществляется динамическая балансировка
- слабая поддержка использования статических свойств, закрытость

# Обзор: система фрагментированного программирования LuNA

Фрагментированное программирование, динамическая балансировка располагают к использованию разнородных вычислительных ресурсов.

Компоненты системы - язык, компилятор, среда исполнения - обеспечивают **возможность работы с динамическими и статическими свойствами, модифицируемость.**

Система LuNA **подходит** для использования в качестве платформы для практических исследований, однако существующая **runtime-система на C++ слабо переносима** и не позволяет в полной мере задействовать разнородные вычислительные ресурсы - **нужно создать переносимую runtime-систему.**

## Цель работы

- *разработка переносимой распределенной среды исполнения, способной задействовать разнородные вычислительные ресурсы, для системы фрагментированного программирования задач численного моделирования LuNA.*

**Требования:** исполнение LuNA-программ, переносимость, масштабируемость, возможность модификации.

## Выбор средств: web-технологии

- Являются производительными и переносимыми (от мобильных телефонов до кластеров)
- Активно развиваются при участии крупнейших компаний
- Имеются перспективы для использования в НРС

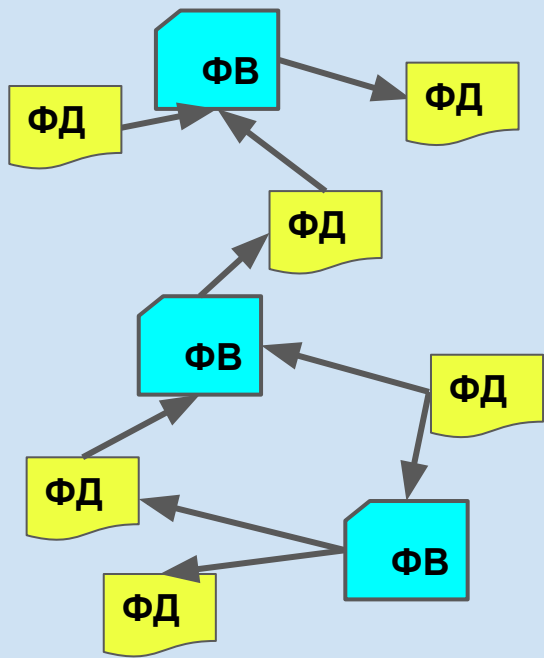


JavaScript

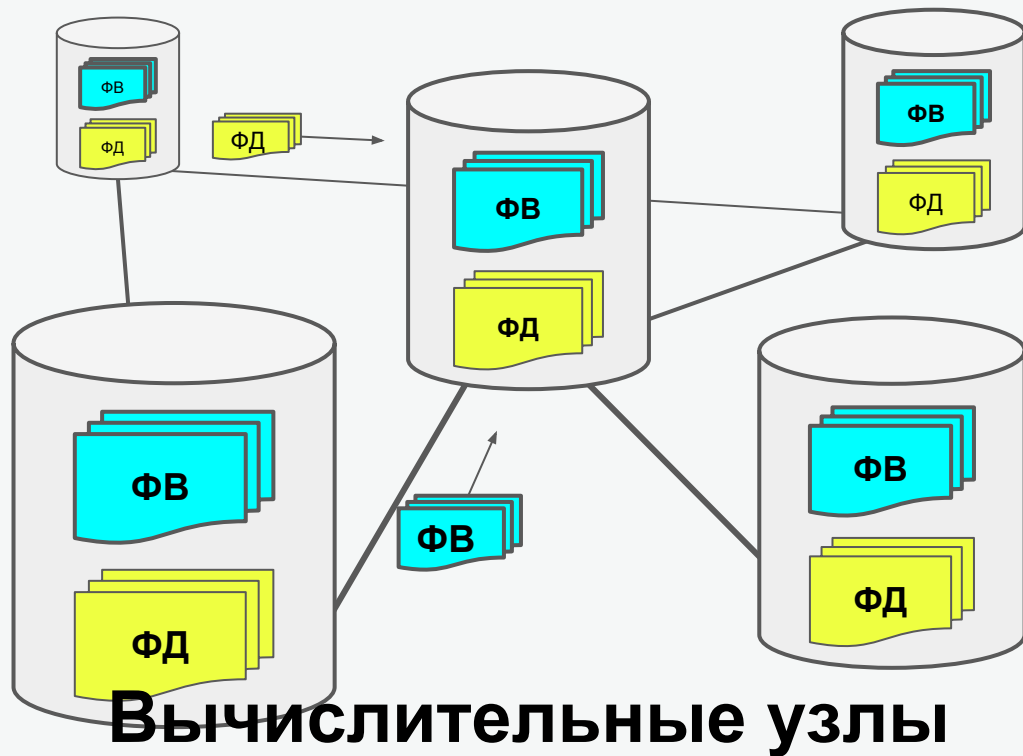
Web  RTC

Plugin-free real time  
communication

# Фрагментированное программирование



Алгоритм



Вычислительные узлы

# Фрагментированное программирование: функционал узла вычислительной сети

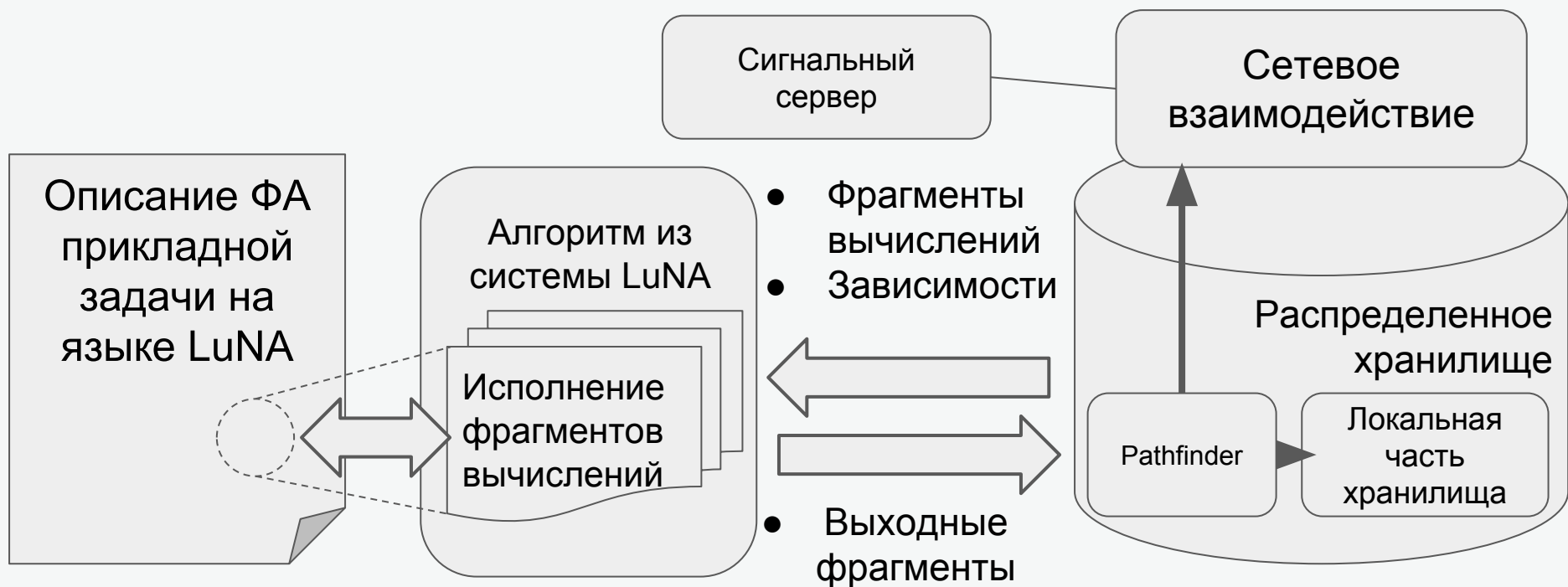
В случае распределенного исполнения на каждом узле вычислительной сети происходит:

- Хранение ФВ и ФД
- Исполнение ФВ
- Ретрансляция информации

При разработке системы исполнения для узла вычислительной сети необходимо обеспечить выполнение данных задач.



Была разработана архитектура системы исполнения для узла вычислительной сети:



# Результат

Local storage	Element view	Node ID: 0	Network view	Processed:																				
Data fragments	Server connection: OPEN		Position: 1 of 3	Processed CFs																				
<table border="1"><thead><tr><th>ID</th><th>Value</th></tr></thead><tbody><tr><td><a href="#">1.main.dummy[1]</a></td><td>1</td></tr><tr><td><a href="#">1.main.dummy[4]</a></td><td>4</td></tr></tbody></table>	ID	Value	<a href="#">1.main.dummy[1]</a>	1	<a href="#">1.main.dummy[4]</a>	4	Left peer: Out of line			<table border="1"><thead><tr><th>ID</th><th>Type</th></tr></thead><tbody><tr><td><a href="#">1.main.1.for[0].1.while[2]</a></td><td>while</td></tr><tr><td><a href="#">1.main.1.for[0].1.while[3].1.[1]</a></td><td>extern</td></tr><tr><td><a href="#">1.main.1.for[0].1.while[4].1.[3]</a></td><td>extern</td></tr><tr><td><a href="#">1.main.1.for[0].1.while[2]</a></td><td>while</td></tr><tr><td><a href="#">1.main.1.for[0].1.while[2]</a></td><td>while</td></tr><tr><td><a href="#">1.main.1.for[0].1.while[2].0.[2]</a></td><td>extern</td></tr></tbody></table>	ID	Type	<a href="#">1.main.1.for[0].1.while[2]</a>	while	<a href="#">1.main.1.for[0].1.while[3].1.[1]</a>	extern	<a href="#">1.main.1.for[0].1.while[4].1.[3]</a>	extern	<a href="#">1.main.1.for[0].1.while[2]</a>	while	<a href="#">1.main.1.for[0].1.while[2]</a>	while	<a href="#">1.main.1.for[0].1.while[2].0.[2]</a>	extern
ID	Value																							
<a href="#">1.main.dummy[1]</a>	1																							
<a href="#">1.main.dummy[4]</a>	4																							
ID	Type																							
<a href="#">1.main.1.for[0].1.while[2]</a>	while																							
<a href="#">1.main.1.for[0].1.while[3].1.[1]</a>	extern																							
<a href="#">1.main.1.for[0].1.while[4].1.[3]</a>	extern																							
<a href="#">1.main.1.for[0].1.while[2]</a>	while																							
<a href="#">1.main.1.for[0].1.while[2]</a>	while																							
<a href="#">1.main.1.for[0].1.while[2].0.[2]</a>	extern																							
Computational fragments	Right peer: open			Console																				
<table border="1"><thead><tr><th>ID</th><th>Type</th></tr></thead></table>	ID	Type	<pre>[INCOMING]##### "Received message from peer:" {   "type": "df",   "df": {     "type": "result",     "localName": null,     "globalName": "1.main.dummy[4]",     "value": 4,     "parent": {       "task": {         "body": [           {             "code": "show",             "rules": [1]          }         ]       }     }   } }</pre>			<table border="1"><tbody><tr><td>C_SHOW: It: 401</td></tr><tr><td>C_SHOW: It: 503</td></tr><tr><td>C_SHOW: It: 302</td></tr></tbody></table>	C_SHOW: It: 401	C_SHOW: It: 503	C_SHOW: It: 302															
ID	Type																							
C_SHOW: It: 401																								
C_SHOW: It: 503																								
C_SHOW: It: 302																								
	<pre>[COMM]##### "Incoming DF from peer 1: " {   "type": "result",   "localName": null,   "globalName": "1.main.dummy[4]",   "value": 4,   "parent": {     "task": {       "body": [         {           "code": "show",           "rules": [1]        }       ]     }   } }</pre>																							

# Результат: тестирование

## Тест:

решение задачи Пуассона методом Якоби в трехмерной области 10x10x10,  
количество фрагментов - 10, значение  $E = 0.0001$

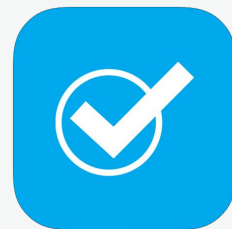
**Топология сети** - “линейка”

**Функция распределения** - хеш-значение от идентификатора фрагмента

---

## Тест интеграции разнородных вычислительных ресурсов:

- Домашний компьютер Windows 7 (Google Chrome)
- Ноутбук Ubuntu 16.04 (Mozilla Firefox)
- 2 x Планшетный компьютер Android (мобильный Google Chrome)



## Тест производительности (домашний компьютер + ноутбук):

1 процесс - 26 сек. | 2 проц. - 356 сек. | 4 проц. - 76 сек. | 8 проц. - 108 сек.

## Заключение

- Разработан алгоритм и архитектура системы распределенного исполнения фрагментированных программ для системы автоматизации ПП LuNA на базе web-технологий.
- Полученная система исполнения позволяет распределенно исполнять параллельные программы на разнородных вычислителях, легко развертывается на широком классе устройств.
- Произведен успешный тестовый запуск - корректное распределенное исполнение на разнородных узлах. Проведены тесты производительности.

Таким образом, разработана платформа для дальнейших экспериментов с выполнением параллельной программы в неоднородной среде.

Спасибо за внимание

Ажбаков Артем Альбертович, ФИТ НГУ, 4 курс