

Использование вычислительных моделей для обеспечения автоматического конструирования фрагментированных программ в системе LuNA

Софронов И.В.

магистрант 2 курса ФИТ НГУ

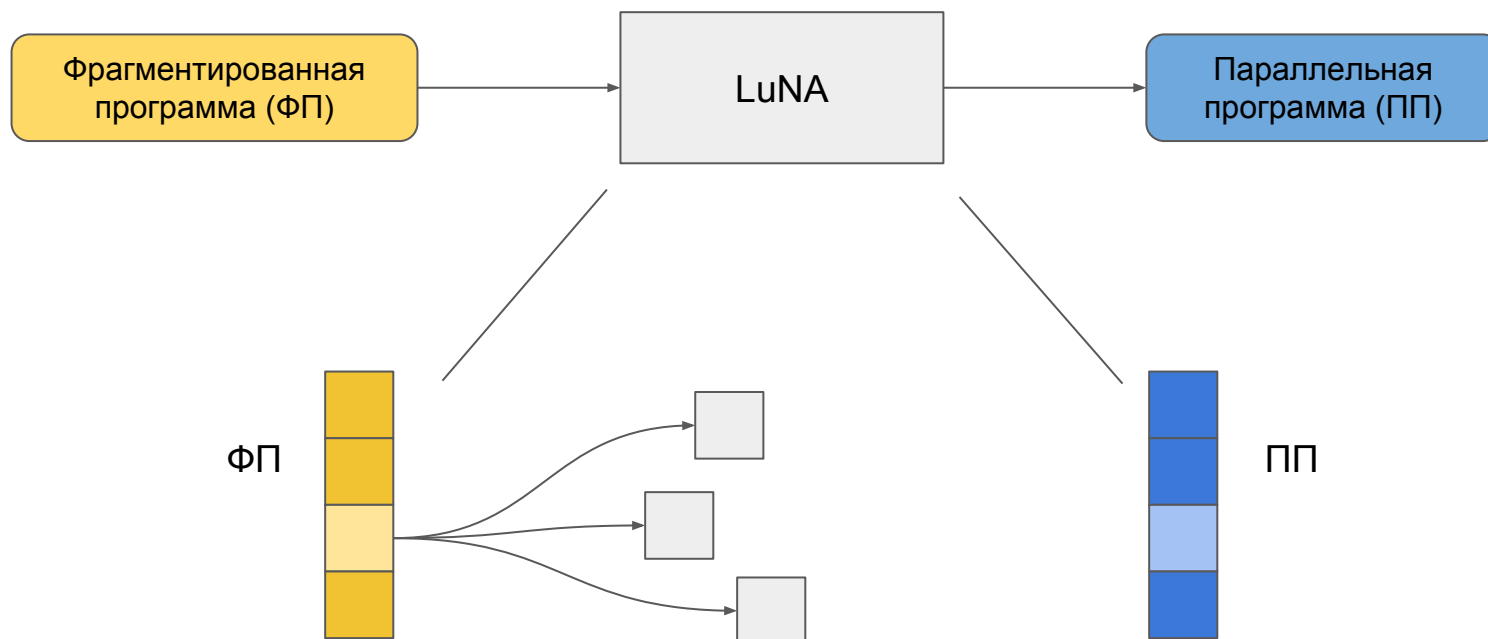
Руководитель Малышкин В.Э.

д.т.н., проф., ИВМиМГ СО РАН

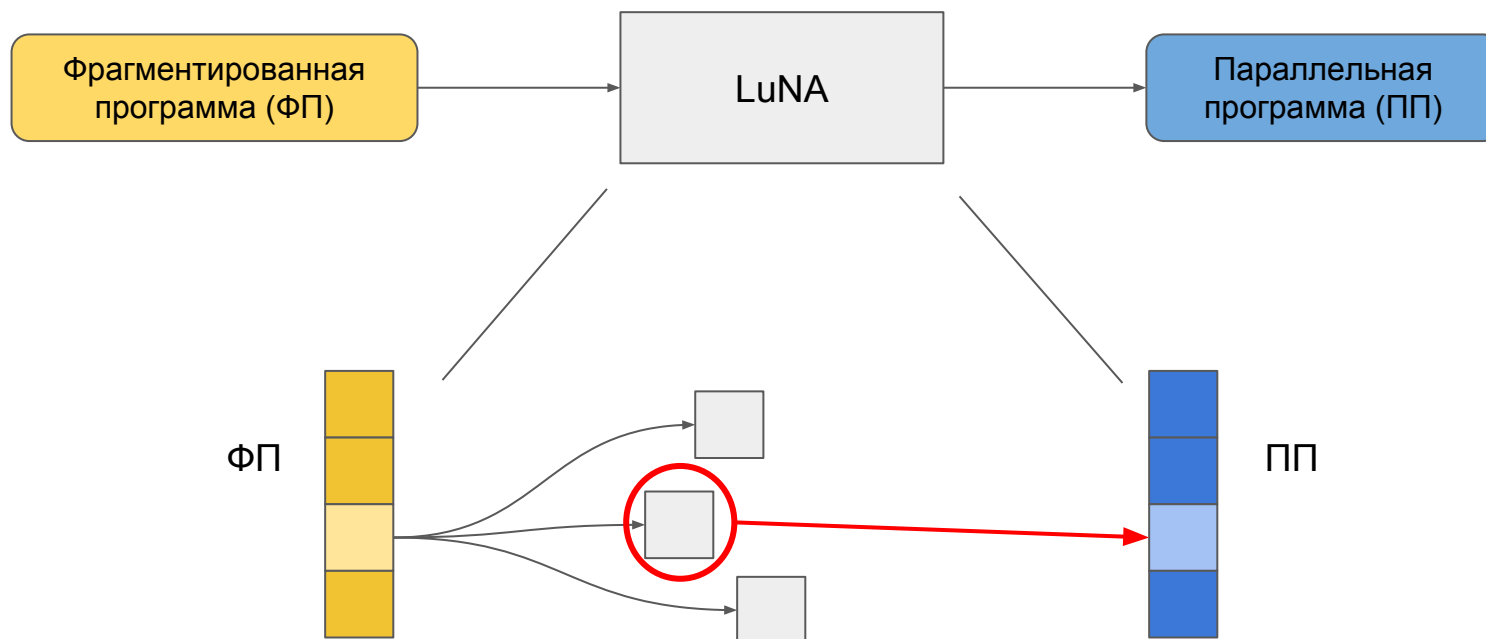
Система фрагментированного программирования (СФП) LuNA

- разрабатывается в ИВМиМГ СО РАН
- предназначена для поддержки параллельной реализации больших численных моделей на суперкомпьютерах
- пользователь задаёт способ разбиения своей задачи на подзадачи, а обеспечение их исполнения ложится на систему

Выбор реализации подзадач в СФП LuNA



Выбор реализации подзадач в СФП LuNA



Проблема выбора модуля для реализации подзадачи

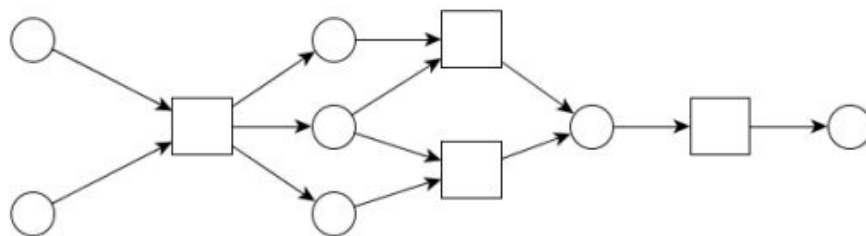
- каждый модуль обладает некоторыми свойствами
- выбор модуля влияет на свойства сгенерированной программы
- поэтому если корректировать программу в соответствии с особенностями вычислителя, то можно получать каждый раз оптимальную программу
- проблема: как, имея требования к программе (критерии) и конфигурацию вычислителя (количество процессов и узлов), выбрать модуль таким образом, чтобы получилась оптимальная, с т.з. заданного критерия, программа?

Подходы к решению проблемы выбора

- фиксированные наборы модулей
- итеративный поиск и генетический алгоритм
- онтологии в связке с алгоритмами кластеризации и машинного обучения
- вычислительные модели

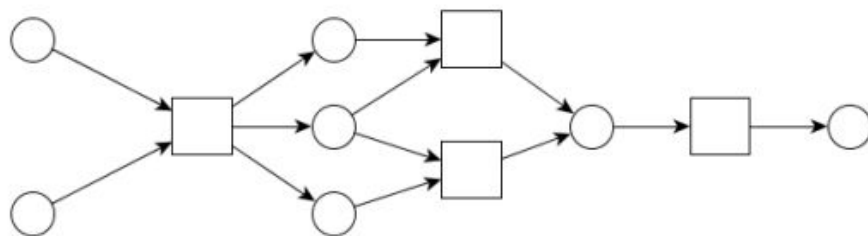
Вычислительные модели (ВМ)

- Описаны в теории синтеза ПП на вычислительных моделях¹
- Двудольный граф, в котором одно подмножество вершин называется “переменные”, другое - “операции”
- Используется для формализации отношений величин в некоторой предметной области



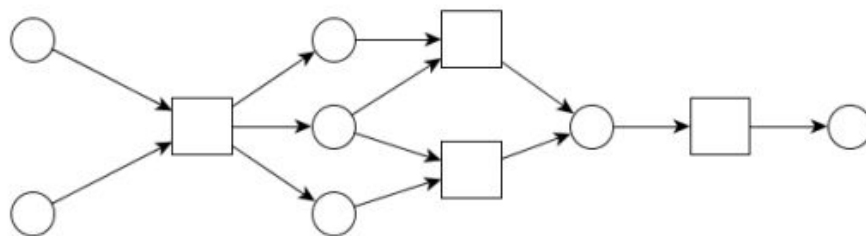
Вычислительные модели (ВМ)

- Задача на ВМ - найти путь из конечного множества исходных переменных в конечное, непересекающееся с исходным, множество результирующих переменных
- Планирование - процесс выделения подграфа в ВМ - плана
- План - решение задачи, поставленной в предметной области, описанной с помощью ВМ



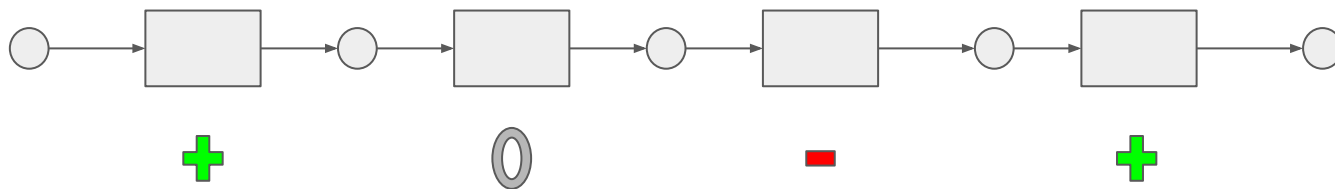
Предлагаемая организация базы на основе VM

- операций в VM - процедура применения модуля: входные переменные этой операции - условия запуска модуля, а выходные - результат его работы
- каждая операция имеет атрибут, который учитывается при выборе модуля в процессе планирования



Использование атрибутов для формирования оценки плана

- атрибут может удовлетворять требованиям, не удовлетворять и не учитываться требованиями;
- за каждую операцию в плане проставляется оценка: 1, -1, 0 соответственно
- сложив все оценки операций в плане получим общую оценку плана
- план с наибольшей оценкой считается наиболее подходящим под указанные требования



Описание VM: операции и переменные

Для описания VM был создан язык LCMD (Language of Computational Model Description) и его интерпретатор

```
model simple {  
variable input, output;  
< attr1, attr2 >  
  { input } -> simple_op1 -> { output } :  
    command 'Run command1' run 'some_prog1' with 'arg1', 'arg2' mod default;  
};
```

В угловых скобках указываются атрибуты операции;

В фигурных скобках - входные/выходные переменные;

После двоеточия располагается секция команд, которые связаны с этой операцией.

Описание VM: задачи

- Одно описание VM может содержать несколько задач
- Каждая задача содержит описание вычислителя (*using*) и критерий (*criteria*)

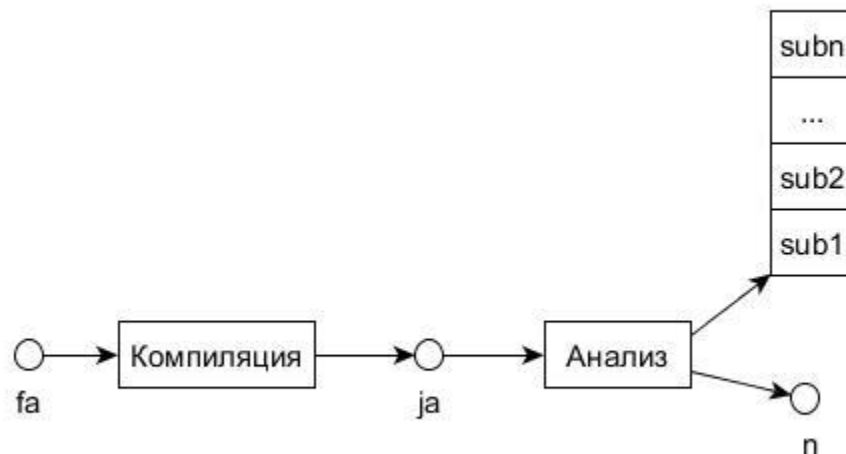
В данном примере конфигурация вычислителя: 3 узла, на каждом из которых запускается 2 процесса из 4 ВОЗМОЖНЫХ

1. **task simple {**
2. **given input;**
3. **required output;**
4. **using 2 of 4 in 3 nodes;**
5. **criteria time_improvement;**
6. **};**

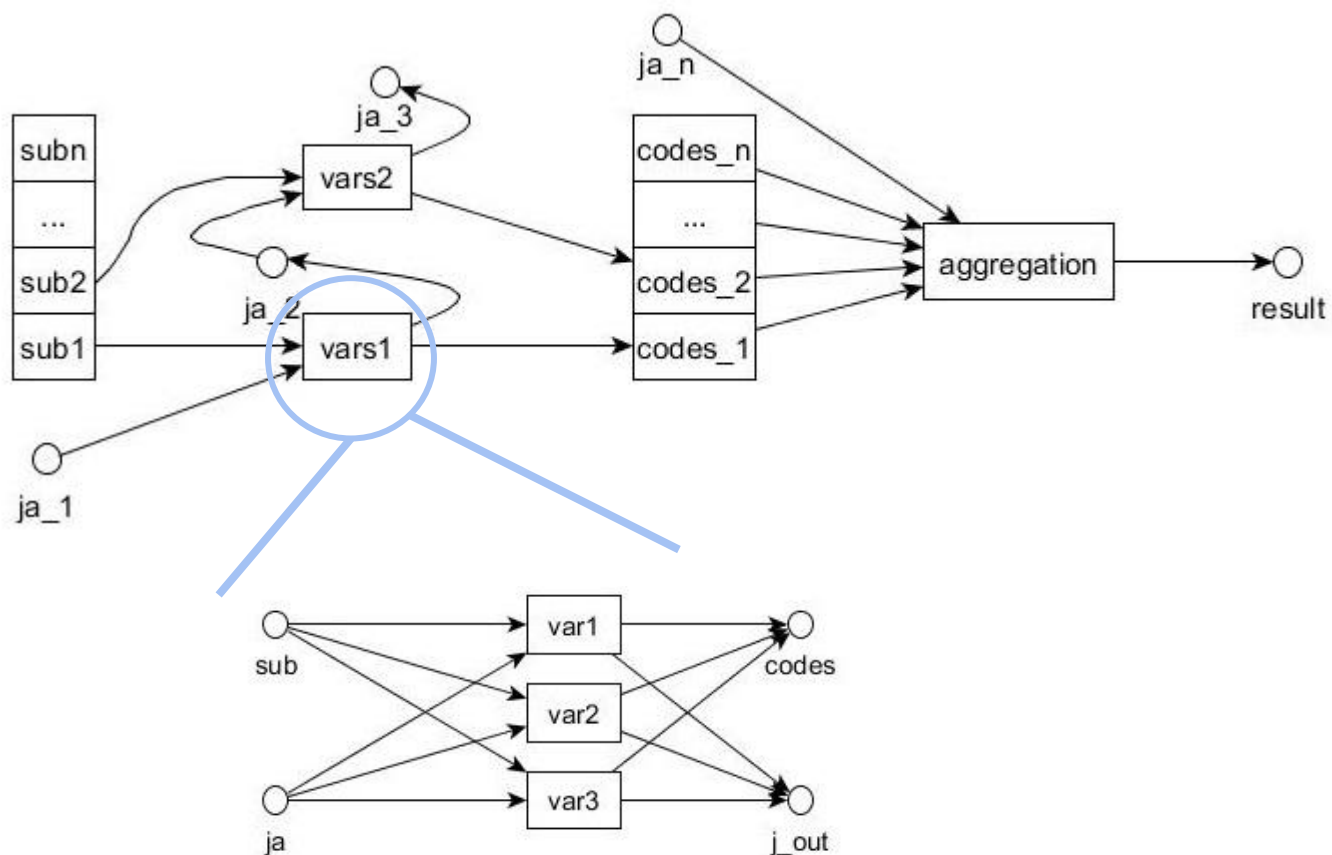
VM конструирования ПП

Содержит три стадии:

- анализ
- выбор варианта
- агрегация

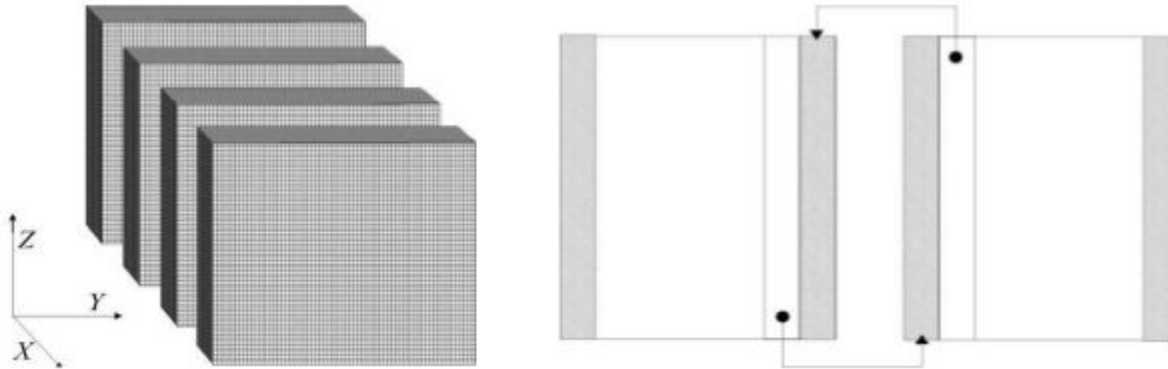


VM конструирования ПП



Тестирование системы

- решение уравнения Пуассона методом Якоби в трёхмерной области при одномерной пространственной декомпозиции



Тестирование системы

Две версии реализации основного цикла расчёта, использующие разные модели управления вычислениями:

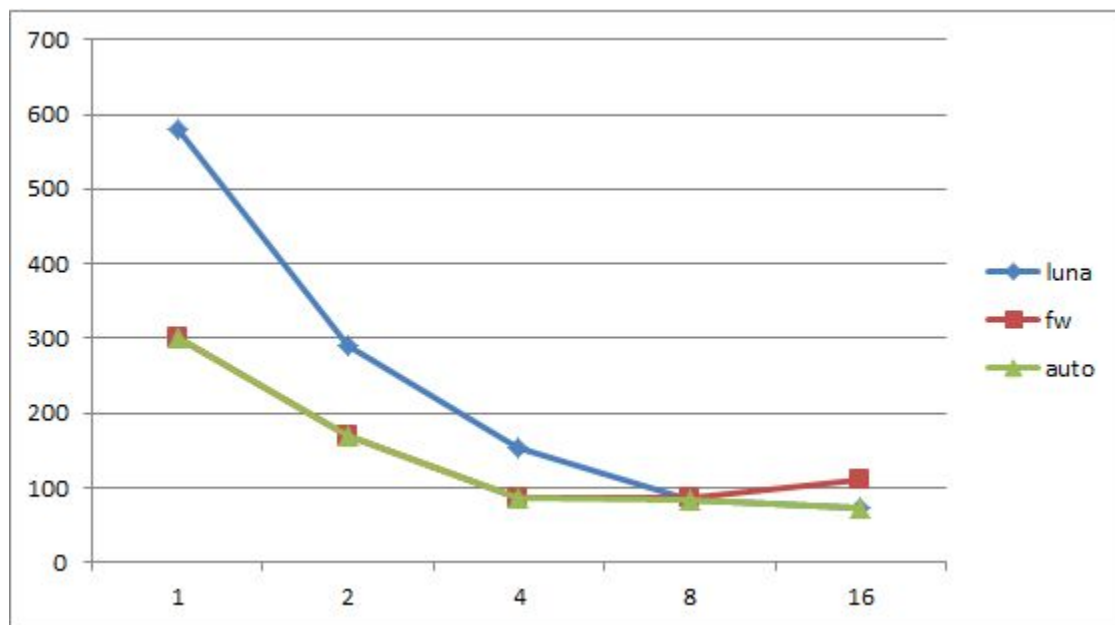
- luna (data-flow)
- fw (event-driven)

Запуск задач производился на кластере МВС-10П (2 процессора Xeon E5-2690, 64 Гб оперативной памяти, коммуникационная сеть на базе FDR Infiniband)

Тестирование системы

Размер области 1200x300x300 (double); 128 итераций; 40 фрагментов; тестирование в общей памяти

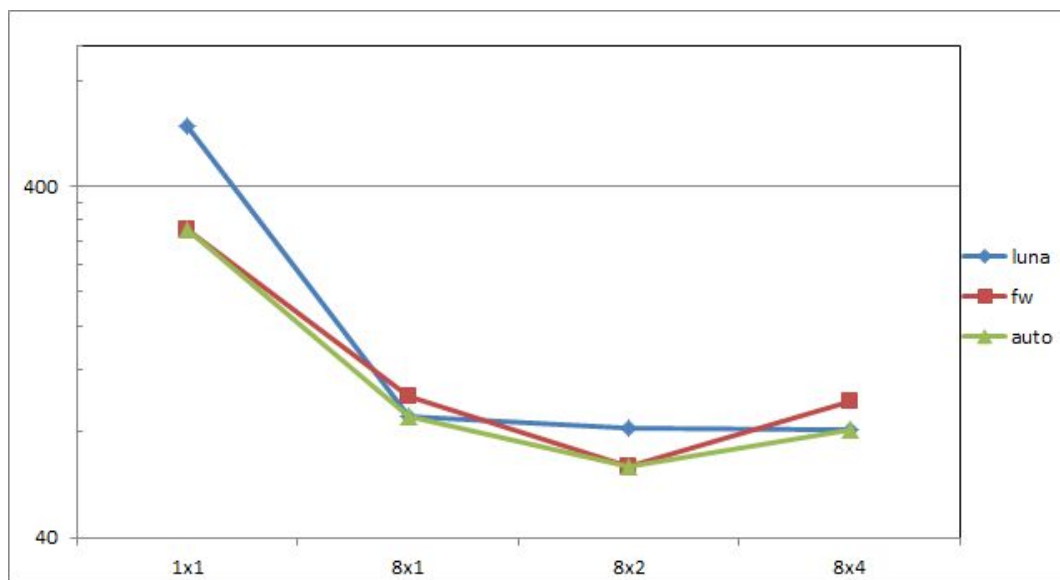
	1	2	4	8	16
luna	581,153	289,234	152,91	83,755	71,716
fw	301,1	168,875	85,55	86,186	109,07



Тестирование системы

Размер области 1200x300x300 (double); 128 итераций; 40 фрагментов;
Запуски отличались количеством процессов (первая цифра в первой строке) и узлов (вторая цифра)

	1x1	8x1	8x2	8x4
luna	591,966	88,143	82,236	81,101
fw	301,103	101,259	63,808	97,749



Заключение

- предложен подход к организации знаний по конструированию программ с использованием вычислительных моделей
- реализована подсистема конструирования ПП, включающая в себя:
 - язык для описания ВМ и его интерпретатор;
 - вычислительные модели, описывающие процесс конструирования и алгоритмы для извлечения знаний из них
 - модули для конструирования ПП
- проведено тестирование, показавшее применимость подхода

Спасибо за внимание!