МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет информационных технологий
Кафедра параллельных вычислений

09.03.01 Информатика и вычислительная техника. Программная инженерия и компьютерные науки

РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМА ПЛАНИРОВАНИЯ НА ВЫЧИСЛИТЕЛЬНЫХ МОДЕЛЯХ С МАССИВАМИ

Выполнил: Морозов Руслан Сергеевич, студент группы 20203

Научный руководитель: Малышкин Виктор Эммануилович, зав. каф. ПВ ФИТ, д.т.н., профессор

Соруководитель: Перепёлкин Владислав Александрович, ст. преп. каф. ПВ ФИТ, к.т.н.

24.06.2024

Новосибирск, 2024

Актуальность проблемы

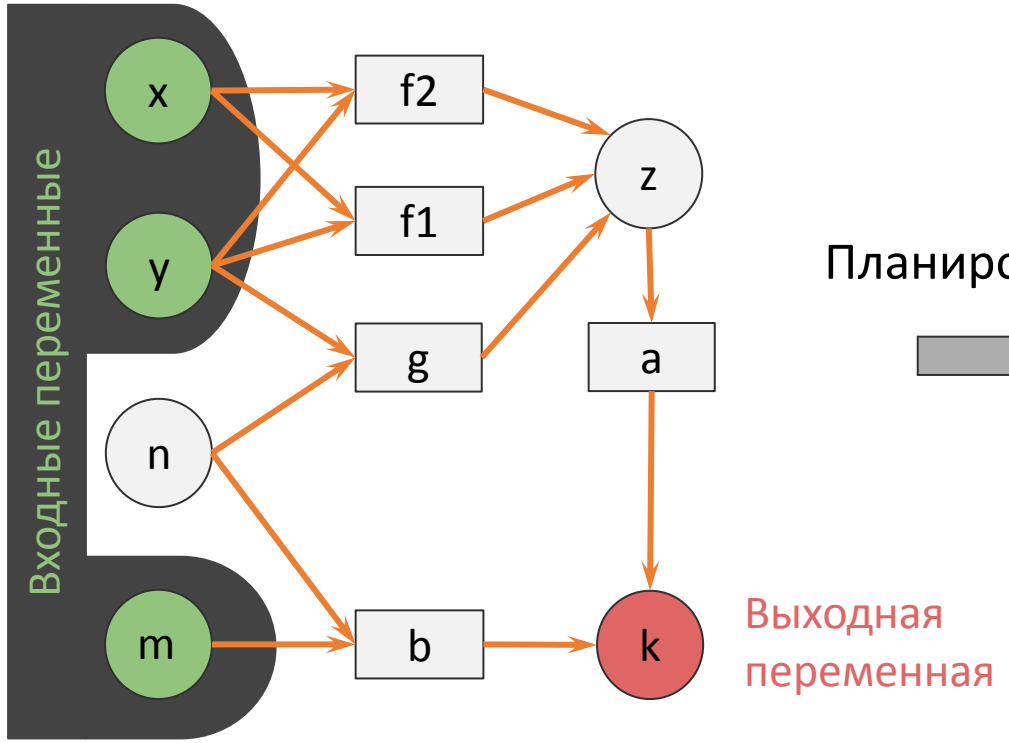
Параллельные программы позволяют увеличить скорость вычислений во многих прикладных задачах численного моделирования.

Разработка параллельных программ – сложная задача, требующая специальных навыков.

Интерес представляет автоматическое конструирование параллельных программ.

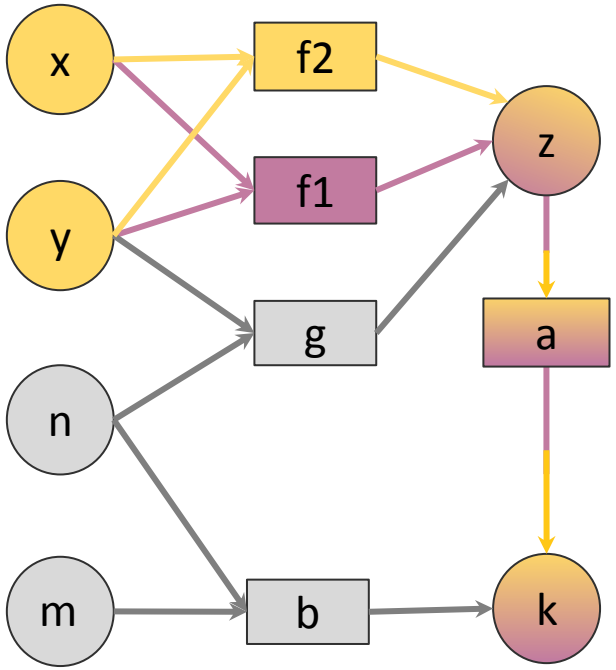
Введение в предметную область

Один из подходов – синтез параллельных программ на основе вычислительных моделей.



Пример задачи на вычислительной модели (ВМ)

Планирование



Возможные планы вычисления

Цель работы и задачи

Целью работы является разработка и реализация алгоритма планирования на вычислительных моделях с массивами (ВММ), применимого для практически значимых задач.

Задачи:

1) Выполнить анализ существующих подходов и решений.
2) Выделить классы практически значимых задач, представимых в виде ВММ.
3) Разработать детализированное описание алгоритма планирования на ВММ.
4) Реализовать систему планирования и разработать форматы описания входных и выходных данных.
5) Выполнить тестирование системы.

Обзор решений

Тыугу Э. Х. Решение задач на вычислительных моделях (1970)

– простые ВМ и планирование через обход в ширину

Карзанов А. В., **Фараджев И. А.** Планирование вычислений при решении задач на вычислительных моделях (1975)

– узкий класс простых ВМ с эффективным планированием

Новосельцев В. Б. Теория структурных функциональных моделей (2006)

– рекурсивные отношения для простых ВМ

Вальковский В. А., **Малышкин В. Э.** Синтез параллельных программ и систем на вычислительных моделях (1988)

– упрощённые ВММ и ВММ со счётчиками и предикатами

Малышкин В. Э., **Панкратов С. А.** Алгоритм имитации планирования (1986)

– алгоритм планирования для упрощённых ВММ

Обзор решений

Перепёлкин В. А. Язык и система LuNA для автоматического конструирования программ численного моделирования

– поддержка ВМ с массивами, но нет возможности планировать.

Парфенов Д. Р., **Мокроусов А. С.** Алгоритм планирования на вычислительных моделях с учётом нефункциональных требований на основе моделирования исполнения

– оптимизация плана для простых ВМ

- Нет реализаций, позволяющих планировать вычисления для ВМ с массивами
- Предложенные способы задания ограничений основательны и достаточно сложны в реализации
- Есть системы для построения эффективных параллельных программ, которые можно использовать вместе с планировщиком на ВММ

Требования к решению

- Возможность описать множества однотипных объектов в виде одномерных массивов
- Поддержка произвольных линейных функций от индекса в массовых операциях

$$f_i\text{: } \texttt{X[2i+1]} \rightarrow \texttt{Y[3i+2]}$$

- Возможность использования для некоторых классов практически значимых задач
- Сложность алгоритма зависит только от сложности описываемых отношений в ВМ
- Находятся <u>все</u> возможные планы (препланирование)

Описание решения

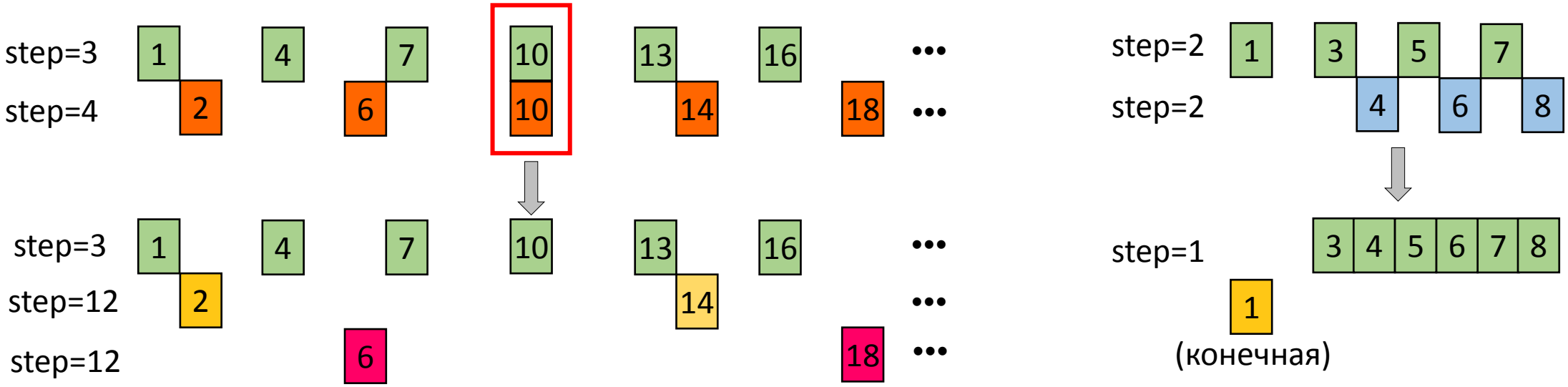
Алгоритм планирования в два этапа:

- На общей задаче ($V_о$,$W_о$) – поиск общих закономерностей
  - Здесь используется Алгоритм имитации планирования (Малышкин В. Э., Панкратов С. А.)
  - Обход в ширину от $V_о$ – сокращается за счёт выделения пси-операций
  - В результате получается ВМ, подобная простой, но аналогичная исходной
  - Обход в ширину в обратном порядке от $W_о$ – отсечение лишних операций
- На частной задаче ($V_ч$,$W_ч$) – определение конкретных диапазонов
  - Обход в ширину от $W_ч$ – вычисление границ для пси-операций
  - Обход в ширину от $V_ч$ – раскрутка пси-операций до исходных вхождений

Описание решения

Разработано представление счётных множеств в виде набора арифметических прогрессий.

Определены правила и приоритеты сокращения до универсального представления.



Алгоритм планирования, ориентированный на программную реализацию, детально описан в тексте ВКР.

Программная реализация

Разработана библиотека для Python

```
modelOperations=[
    MOperation('f', ins=[('x',(1,0))], outs=[('y',(1,0))], VS([('f',[R((1,0))])])),
    MOperation('g', ins=[('y',(1,0))], outs=[('x',(1,1))], VS([('g',[R((1,0))])])),
    MOperation('a', ins=[('y',(1,0))], outs=[('z',(1,0))], VS([('a',[R((1,0))])]))
    ]
modelVariables=[
    MVariable('x', entry=['f'], comp=['e','g']),
    MVariable('y', entry=['g','a'], comp=['f']),
    MVariable('z', entry=[], comp=['a’])
    ]
simplemodel = Model(v={x.label : x for x in modelVariables},
                o={x.label : x for x in modelOperations})
simplemodel_general = model_plan(simplemodel,
            V=VS([('x',[R(1)])]),
            W=VS([('x',[R((1,2))]), ('z',[R((1,1))]), ('y',[R((1,1))])]))
simplemodel_specific = model_fit(simplemodel, simplemodel_general,
            V=VS([('x',[R(1)])]),
            W=VS([('x',[R(10,31,2)]),('z',[R(100)])]))
```

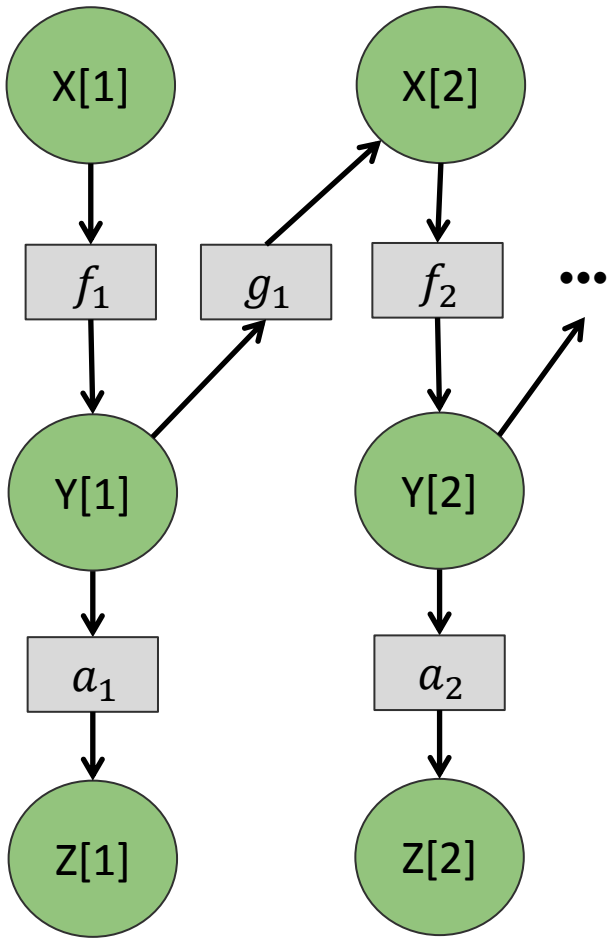
Операции

Переменные

ВМ/ВММ

Общая задача

Частная задача



Программная реализация

Разработан JSON-формат описания ВММ.

Разработан серверный компонент с использованием фреймворка Tornado, предоставляющий REST API планирования на простых ВМ и ВММ.

```
{
  "name": "пример ВММ",
  "variables": [
    { "id": "n", "type": "simple"  },
    { "id": "x", "type": "massive" },
    { "id": "y", "type": "massive" },
  ],
  "operations": {
    "f": {
      "type": "massive",
      "scope": [[2, 5, 6]],
      "description": "вычисляет (x[i], n) -> y[2*i+1]",
      "inputs": [
        { "id": "n" },
        { "id": "x", "scope": [1, 0] }
      ],
      "outputs": [
        { "id": "y", "scope": [2, 1] }
      ],
    }
  }
}
```

$i \in \{5,7,9,11,13\}$

Анализ решения

Преимущества:

- Некоторые классы задач удобно описывать предложенным образом
- Возможность программной реализации алгоритма

Недостатки:

- Предложенный способ представления множеств не всегда работает эффективно, трудно расширяем до многомерного случая
- Некоторые классы задач не представимы в виде общей задачи на ВММ

Тестирование

Проверялась возможность планирования и его результат, удобство описания вычислительной модели и задач.

Рассмотрены практически значимые задачи:

- Вычисление сейсмической свёртки по файлу сеанса, опорному сигналу и интервалу дискретизации. Два способа вычисления (на CPU и GPU)
- Вычисление состояний клеточно-автоматной модели потока жидкости FHP-MP и их визуализаций

Достигнуты ожидаемые результаты для всех тестовых задач.

Описание задачи сейсмической свёртки

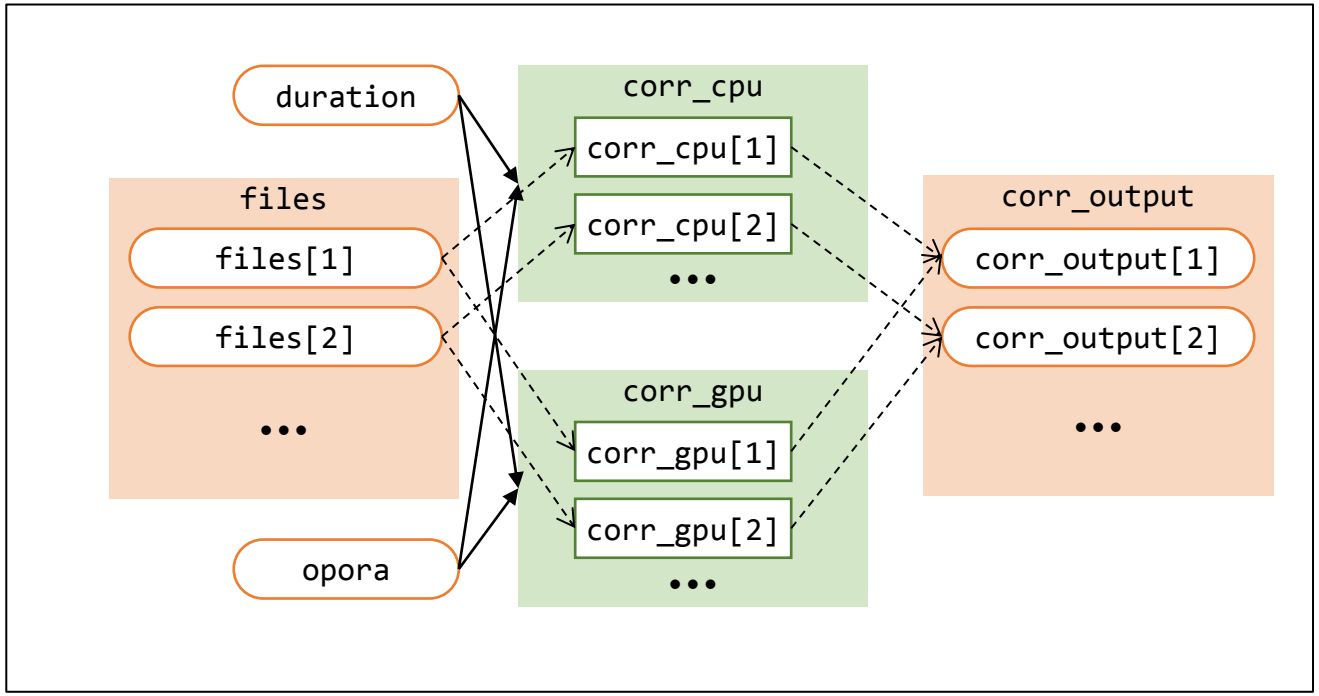
```
modelOperations=[
    MOperation('corr_cpu',
               ins=[('files',(1,0)), 'opora', 'duration'],
               outs=[('corr_output',(1,0))],
               scope=VS([('corr_cpu',[R((1,0))])])
               ),
    MOperation('corr_gpu',
               ins=[('files',(1,0)), 'opora', 'duration'],
               outs=[('corr_output',(1,0))],
               scope=VS([('corr_gpu',[R((1,0))])])
               )
]

modelVariables=[
    SVariable('opora', entry=['corr_cpu','corr_gpu'], comp=[]),
    SVariable('duration', entry=['corr_cpu','corr_gpu'], comp=[]),
    MVariable('corr_output', entry=[], comp=['corr_cpu','corr_gpu']),
    MVariable('files', entry=['corr_cpu','corr_gpu'], comp=[])
]

seis_model = Model(v={x.label : x for x in modelVariables},
              o={x.label : x for x in modelOperations})

# планирование на общей задаче
seis_model_t1 = model_plan(seis_model,
                           V=VS([('files',[R((1,0))]), 'opora', 'duration']),
                           W=VS([('corr_output',[R((1,0))])])
                           )
# планирование на частной задаче
seis_model_t2 = model_fit(seis_model, seis_model_t1,
                          V=VS([('files',[R(0,1000)]), 'opora', 'duration']),
                          W=VS([('corr_output',[R(0,100)])])
                          )
```

Использование библиотеки для планирования



Вычислительная модель задачи

Полученный результат

```
{
    'opora': opora(entry:['psi0', 'psi1'], comp:[]),
    'duration': duration(entry:['psi0', 'psi1'], comp:[]),
    'corr_output': corr_output(entry:[], comp:['psi0', 'psi1']),
    'files': files(entry:['psi0', 'psi1'], comp:[])
}
psi0: psi0(psi(auto): corr_gpu[(1, 0)](files[(1, 0)], opora, duration))(
    ins:{
        duration; opora;
        files: Range(1,0)
    },
    outs:{
        corr_output: Range(1,0)
    })
psi1: psi1(psi(auto): corr_cpu[(1, 0)](files[(1, 0)], opora, duration))(
    ins:{
        duration; opora;
        files: Range(1,0)
    },
    outs:{
        corr_output: Range(1,0)
    })
```

```
{
    'opora': opora(entry:['corr_cpu', 'corr_gpu'], comp:[]),
    'duration': duration(entry:['corr_cpu', 'corr_gpu'], comp:[]),
    'corr_output': corr_output(entry:[], comp:['corr_cpu', 'corr_gpu']),
    'files': files(entry:['corr_cpu', 'corr_gpu'], comp:[])
}
corr_cpu: corr_cpu[{
    corr_cpu: {0,1,...,99}(step=1)
}](
    ins:[('files', (1, 0)), 'opora', 'duration'],
    outs:[('corr_output', (1, 0))]
  )

corr_gpu: corr_gpu[{
    corr_gpu: {0,1,...,99}(step=1)
}](
    ins:[('files', (1, 0)), 'opora', 'duration'],
    outs:[('corr_output', (1, 0))]
  )
```

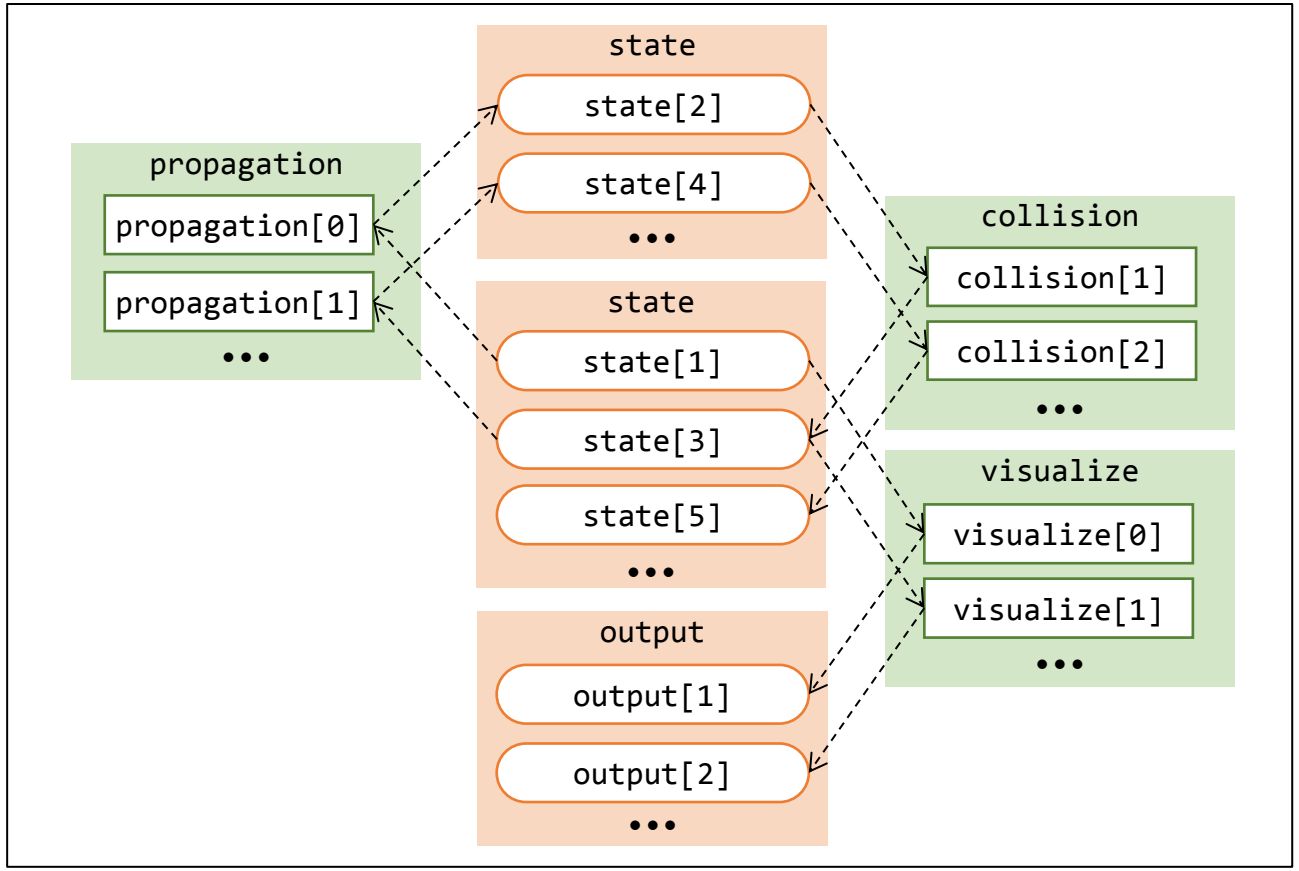
Вывод результатов планирования для общей задачи (слева) и частной задачи (справа).

Жёлтым выделено содержимое пси-операций и итоговые диапазоны массовых операций

Описание задачи с клеточным автоматом

```
1  modelOperations=[
2      MOperation('propagation',
3                 ins=[('state',(2,1))],
4                 outs=[('state',(2,2))],
5                 scope=VS([('propagation',[R((1,0))])])
6                 ),
7      MOperation('collision',
8                 ins=[('state',(2,0))],
9                 outs=[('state',(2,1))],
10                scope=VS([('collision',[R((1,0))])])
11                ),
12     MOperation('visualize',
13                ins=[('state',(2,1))],
14                outs=[('output',(1,1))],
15                scope=VS([('visualize',[R((1,0))])])
16                )
17 ]
18 modelVariables=[
19     MVariable('state',
20               entry=['propagation', 'collision', 'visualize'],
21               comp=['propagation', 'collision']
22               ),
23     MVariable('output',
24               entry=[], comp=['visualize']
25               )
26 ]
27 fhpmp_model = Model(v={x.label : x for x in modelVariables},
28               o={x.label : x for x in modelOperations})
29
30 # планирование на общей задаче
31 fhpmp_model_general = model_plan(fhpmp_model,
32                                 V=VS([('state',[R(1)])]),
33                                 W=VS([('output',[R((1,1))])])
34                                 )
35 # планирование на частной задаче
36 fhpmp_model_specific = model_fit(fhpmp_model, fhpmp_model_general,
37                                 V=VS([('state',[R(1)])]),
38                                 W=VS([('output',[R(20000)])])
39                                 )
```

Использование библиотеки для планирования



Вычислительная модель задачи

Полученный результат

```
{
    'state': state(entry:['visualize', 'psi0', 'psi1'], comp:['psi0']),
    'output': output(entry:[], comp:['visualize', 'psi1'])
}
visualize: visualize[{
    visualize: {0,1}
}](
    ins:[('state', (2, 1))],
    outs:[('output', (1, 1))]
  )

psi0: psi0(psi: collision[(1, 1)](propagation[(1, 0)](state[(2, 1)])))(
    ins:{
        state:  {1}
    },
    outs:{
        state: Range(1,2)
    })

psi1: psi1(psi(auto): visualize[(1, 2)](state[(2, 5)]))(
    ins:{
        state: Range(2,5)
    },
    outs:{
        output: Range(1,3)
    })
```

```
{
    'state': state(
        entry:['propagation', 'collision', 'visualize'],
        comp:['propagation', 'collision']
        ),
    'output': output(entry:[], comp:['visualize'])
}
propagation: propagation[{
    propagation: {0,1,...,19998}(step=1)
}](
    ins:[('state', (2, 1))],
    outs:[('state', (2, 2))]
  )

collision: collision[{
    collision: {1,2,...,19999}(step=1)
}](
    ins:[('state', (2, 0))],
    outs:[('state', (2, 1))]
  )

visualize: visualize[{
    visualize: {19999}
}](
    ins:[('state', (2, 1))],
    outs:[('output', (1, 1))]
  )
```

Вывод результатов планирования для общей задачи (слева) и частной задачи (справа).

Жёлтым выделено содержимое пси-операций и итоговые диапазоны массовых операций

Апробация

Публикации тезисов (с выступлением на конференциях):

- Морозов Р. С. **Реализация алгоритма планирования на вычислительных моделях с одномерными массивами с линейной индексацией** / Всероссийская летняя XLI молодежная Школа-конференция по параллельному программированию с международным участием : Тез. докл – Новосибирск. – 3-14 июля 2023 г. – Новосиб. гос. ун-т. – Новосибирск : ИПЦ НГУ, 2024
- Морозов Р. С. **Реализация алгоритма планирования на вычислительных моделях с одномерными массивами с линейной индексацией** / Марчуковские научные чтения-2023: Тезисы Междунар. конф. – Новосибирск – 18-20 декабря 2023 г. – Ин-т вычислит. математики и матем. геофизики СО РАН, 2024
- Морозов Р. С., Парфенов Д. Р. **Реализация системы оптимизирующего планирования на вычислительных моделях с одномерными массивами** / Материалы 62-й Международной научной студенческой конференции. – Новосибирск. – 17-23 апреля 2024 г. – Новосиб. гос. ун-т. – Новосибирск : ИПЦ НГУ, 2024

Заключение

Защищаемые положения:

- Модификация алгоритма планирования на ВММ и её детализированное описание, рассчитанное на программную реализацию
- Подход к представлению ВММ и формат описания ВММ и задач на ВММ
- Программная реализация модуля планирования, состоящего из библиотеки планирования и серверного компонента

Дальнейшие планы:

- Добавить поддержку многомерных массивов
- Исследовать практическую применимость иных способов задания ограничений вычислений для массовых операций (предикаты и счётчики)
- Разработать способы описания других классов задач

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет информационных технологий
Кафедра параллельных вычислений

09.03.01 Информатика и вычислительная техника. Программная инженерия и компьютерные науки

Спасибо за внимание!

РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМА ПЛАНИРОВАНИЯ НА ВЫЧИСЛИТЕЛЬНЫХ МОДЕЛЯХ С МАССИВАМИ

Выполнил: Морозов Руслан Сергеевич, студент группы 20203

Научный руководитель: Малышкин Виктор Эммануилович, зав. каф. ПВ ФИТ, д.т.н., профессор

Соруководитель: Перепёлкин Владислав Александрович, ст. преп. каф. ПВ ФИТ, к.т.н.

24.06.2024

Новосибирск, 2024

Научная новизна и практическая ценность

Научная новизна работы заключается в детализации существующих алгоритмов планирования на ВММ с разработкой способа представления счётных множеств в памяти ЭВМ, а также в разработке упрощённого формата описания ВММ, практически подкреплённого алгоритмом планирования.

Практическая ценность работы состоит в том, что разрабатываемая система может участвовать в автоматическом конструировании параллельных программ для актуальных задач численного моделирования по относительно простому их описанию.

Приоритеты правил представления множеств

1) Неограниченные диапазоны при возможности заменяются диапазонами с меньшим шагом.
2) Неограниченный диапазон включает в себя максимальное количество элементов.
3) Ограниченные диапазоны не должны пересекаться.
4) Каждый элемент образует последовательность с ближайшим элементом, если для того нет более близкого элемента.

Структура библиотеки планирования

Файл	Размер, строк	Описание
alglogger.py	34	Работа с протоколированием в ходе работы.
base_math.py	32	Вычисление НОК, НОД. Вариант расширенного алгоритма Евклида.
fit_tools.py	231	Вспомогательные процедуры при планировании на частной задаче.
json_fields_names.py	31	Описание полей JSON-формата ВМ.
model.py	238	Описание классов объектов ВМ.
plan.py	238	Основные (высокоуровневые) процедуры планирования.
plan_tools.py	287	Вспомогательные процедуры при планировании на общей задаче.
ranges.py	246	Класс диапазонов.
terms.py	82	Работа с термами при построении пси-операций.
variables_set.py	638	Классы множеств для хранения диапазонов.

Список модулей библиотеки планирования

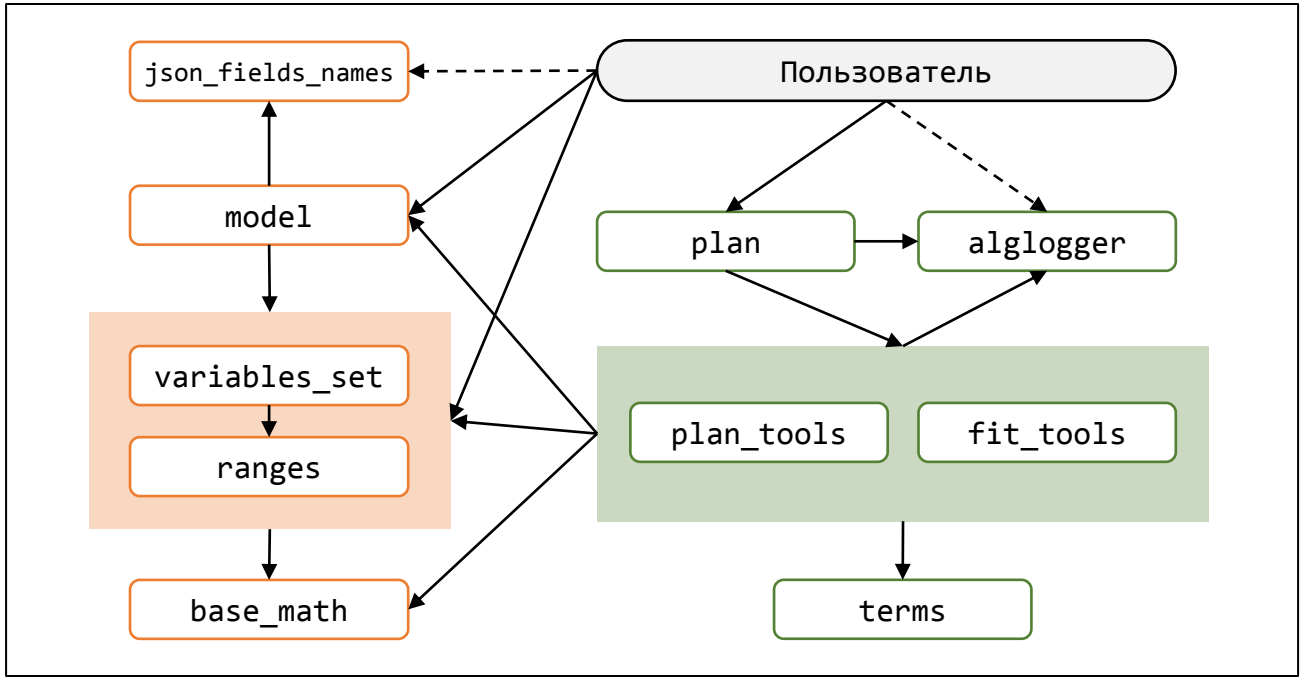


Схема взаимодействия модулей библиотеки планирования