

Реализация алгоритма планирования на вычислительных моделях с массивами

Докладчик: Морозов Р. С.

Руководитель: Перепёлкин В. А.

Всероссийская летняя XLI молодежная Школа-конференция
по параллельному программированию

Новосибирск, 2023

Проблема

- Известная проблема - автоматическое конструирование параллельных программ.
- Один из подходов к решению - вычислительные модели. Они позволяют описать предметную область на низком уровне, поэтому там легко определить зависимости по данным для эффективного параллельного исполнения.
- Задача сводится к выводу программы по описанию потенциально бесконечного множества.

Цель

Подготовить алгоритм планирования на вычислительных моделях с массивами для интегрирования в систему LuNA.

Задачи

1. Исправить существующие ошибки в реализации алгоритма.
2. Реализовать алгоритм трансляции внутреннего представления спланированной модели в код на языке LuNA.
3. Проверить корректность работы алгоритмов на разных примерах входных данных, включая практически значимую задачу.
4. Выполнить оценку производительности алгоритмов.

Вычислительные модели

- Конечное множество переменных
- Конечное множество операций

Вычислительные модели с массивами (упрощённые)

- Добавляется конечное множество массивов (одномерных)
- Добавляется конечное множество массовых операций
- Индексы операций определены на линейных множествах

Вычисление на модели

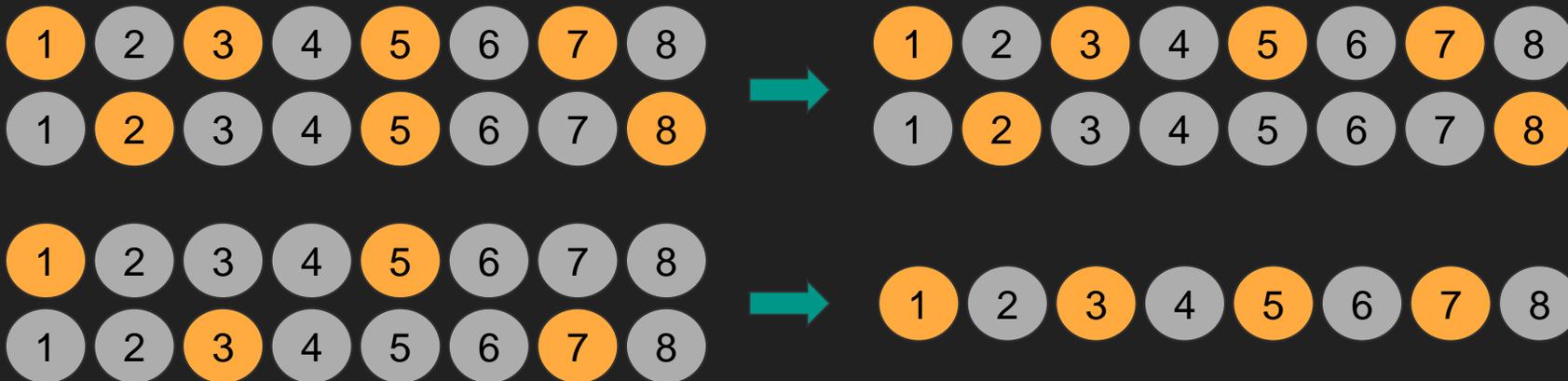
- Задаются множество входных переменных и множество выходных переменных

Идея алгоритма планирования

- Последовательно вычисляются новые переменные.
- При применении массовых операций находятся закономерности — повторяющиеся наборы вхождений массовых операций, такие, что каждый набор не использует переменных, не вычисленных ранее или предыдущими наборами.
- Создаются новые простые операции, у которых вход - конечное число переменных, а выход - бесконечное множество переменных.
- В итоге остаётся только конечное число вхождений массовых операций, для которых не нашлось закономерностей.

Представление бесконечных множеств

- Нужно хранить арифметические прогрессии с различными разностями
- Не должно быть дублирования элементов
- Представление должно быть наиболее простым



Генерация программ

- Отдельный блок кода
- Объявление переменных вычислительной модели
- Разные типы сконструированных операций требуют разные реализации
- Построенные на этапе планирования термы описывают зависимости, а не код вычисления.

Пример

Операции:

init: $\text{init}[i] \rightarrow x[i]; \{0, 1, \dots\} \setminus \{N, N+1, \dots\}$

blurleft: $x[i-N], x[i+1-N] \rightarrow x[i]; \{N, 2N, \dots\}$

blurrightright: $x[i-1], x[i] \rightarrow x[i+N]; \{N-1, 2N-1, \dots\}$

blurmiddle: $x[i-1], x[i], x[i+1] \rightarrow x[i+N]; \{0, 1, \dots\} \setminus (\{0, N, \dots\} \cup \{N-1, 2N-1, \dots\})$

sum0: $x[i] \rightarrow s[i]; \{0, N, \dots\}$

sum: $x[i], s[i-1] \rightarrow s[i]; \{0, 1, \dots\} \setminus \{0, N, \dots\}$

diff: $s[N*i+N-1], s[N*i-1] \rightarrow d[i+1]; \{1, 2, \dots\}$

Пример

```
1 modelOperations=[
2     MOperation('init', [('init',(1,0))] , [('x',(1,0))], VariablesSet([('init',(1,GAUSS_N))])),
3     MOperation('blurleft', [('x',(1,-GAUSS_N)),('x',(1,1-GAUSS_N))] , [('x',(1,0))], VariablesSet([('blurleft',(GAUSS_N,GAUSS_N))])),
4     MOperation('blurrightright', [('x',(1,-1)),('x',(1,0))] , [('x',(1,GAUSS_N))], VariablesSet([('blurrightright',(GAUSS_N,GAUSS_N-1))])),
5     MOperation('blurmmiddle', [('x',(1,-1)),('x',(1,0)),('x',(1,1))] , [('x',(1,GAUSS_N))],
6     | | | VariablesSet([('blurmmiddle',(1,0))].difference(VariablesSet([('blurmmiddle',(GAUSS_N,0))]).union(VariablesSet([('blurmmiddle',(GAUSS_N,GAUSS_N-1))])),
7     MOperation('sum0', [('x',(1,0))] , [('s',(1,0))], VariablesSet([('sum0',(GAUSS_N,0))])),
8     MOperation('sum', [('x',(1,0)),('s',(1,-1))] , [('s',(1,0))],
9     | | | VariablesSet([('sum',(1,0))].difference(VariablesSet([('sum',(GAUSS_N,0))])),
10    MOperation('diff', [('s',(GAUSS_N,GAUSS_N-1)),('s',(GAUSS_N,-1))] , [('d',(1,1))], VariablesSet([('diff',(1,1))])),
11    SOperation('getsn', ['n', ('s',(GAUSS_N,GAUSS_N-1))] , ['sn']),
12    SOperation('getdn', ['n', ('d',(1,2))] , ['dn'])
13 ]
14 modelVariables=[
15     MVariable('init', ['init'], []),
16     SVariable('n', ['getsn'], []),
17     SVariable('sn', [], ['getsn']),
18     SVariable('dn', [], ['getdn']),
19     MVariable('x', ['blurleft','blurrightright','blurmmiddle','sum0','sum'], ['blurleft','blurrightright','blurmmiddle','init']),
20     MVariable('s', ['sum','sum0','getsn','diff'], ['sum','sum0']),
21     MVariable('d', ['getdn'], ['diff'])
22 ]
23 gaussmodel = Model(v={x.label : x for x in modelVariables},
24 | | | o={x.label : x for x in modelOperations})
25
```

Пример

Вычисление $d[15]$ из $init[0..N]$ при $N = 5$

```
{
var x[], s[];
init(0, x[0]);
init(1, x[1]);
init(2, x[2]);
init(3, x[3]);
init(4, x[4]);
sum0(x[70],s[70]);
sum(x[71],s[70],s[71]);
sum(x[72],s[71],s[72]);
sum(x[73],s[72],s[73]);
sum(x[74],s[73],s[74]);
sum0(x[65],s[65]);
sum(x[66],s[65],s[66]);
sum(x[67],s[66],s[67]);
sum(x[68],s[67],s[68]);
sum(x[69],s[68],s[69]);
diff(s[74],s[69],d[15]);
for i = 0..13
{
blurleft(x[5*i+0],x[5*i+1],x[5*i+5]);
blurmiddle(x[5*i+0],x[5*i+1],x[5*i+2],x[5*i+6]);
blurmiddle(x[5*i+1],x[5*i+2],x[5*i+3],x[5*i+7]);
blurmiddle(x[5*i+2],x[5*i+3],x[5*i+4],x[5*i+8]);
blurrigh(x[5*i+3],x[5*i+4],x[5*i+9]);
}
}
```

Ограничения

- Генерация кода только для вычисления конкретных значений, без ссылочной адресации.
- Невозможность повторяющихся вычислений переменных в системе LuNA.
- Работа только с одномерными массивами
- Производительность

Заключение

- Расширен класс для хранения бесконечных множеств, соответствующий предложенным требованиям.
- Написан алгоритм генерации LuNA кода, корректно работающий для некоторого класса задач.
- Получены оценки производительности и выявлены узкие места алгоритма, которые в дальнейшем будут улучшены.

Планы

- Создать язык описания программ с вычислительными моделями на основе языка LuNA и компилятор для него, использующий полученные алгоритмы
- Добавить возможность использовать многомерные массивы.
- Добавить возможность генерации кода при ссылочной адресации.