Новосибирский Государственный Университет
Факультет информационных технологий
Высокопроизводительные вычислительные системы

Разработка моделей и средств визуального фрагментированного программирования численных алгоритмов

Инга Пясс, гр.13221

План

- 1. Постановка задачи
- 2. Обзор
  - 2.1 Подходы к параллельному программированию
  - 2.2 Языки визуального параллельного программирования
  - 2.3 CODE and HeNCE
  - 2.4 VPE
- 3. Модель Visual LuNA++
  - 3.1 Операторная схема
  - 3.2 Стандартная модель
  - 3.3 Модель Visual LuNA++
- 4. Планы
  - 4.1 Планы на следующий семестр
  - 4.2 Планы на второй курс

Постановка задачи

- Разработка модели языка визуального параллельного программирования для записи численных алгоритмов
- Разработка способов графического отображения алгоритмов на визуальном языке
- Доработка модели визуального языка и средств отображения для представления фрагментированных алгоритмов
- Разработка и реализация системы визуального фрагментированного программирования

Подходы к параллельному программированию

Наиболее распространены подходы к параллельному программированию, идущие от последовательного программирования:

- Автоматическое выявление параллелизма компиляторами в последовательных программах (распараллеливающие компиляторы).
- Обозначение в последовательных программах независимых вычислений.
- Использование в последовательных языках явных операций передачи данных и параллельного запуска задач.

Языки визуального параллельного программирования

1) **языки на основе представления зависимостей по данным и управлению (control and data flow)**, в которых графические узлы представляют фрагменты вычислений, и дуги представляют зависимости между ними,

2) **коммуникационные языки (communication model languages),** в которых графические узлы представляют собой процессы, в то время как дуги представляют каналы передачи сообщений между ними.

CODE and HeNCE

Графы для представления параллельных программ – узлы представляют собой фрагменты вычислений, а дуги представляют собой зависимости между ними.

Преимущества:

1) Модели программирования обоих языков являются высокоуровневыми, что освобождает программиста от необходимости рассматривать низкоуровневые аспекты исполнения (например, кодирование операций синхронизации).

2) Выявлен неявный параллелизм, так как программисту необходимо только указать зависимости между фрагментами вычислений в форме графа.

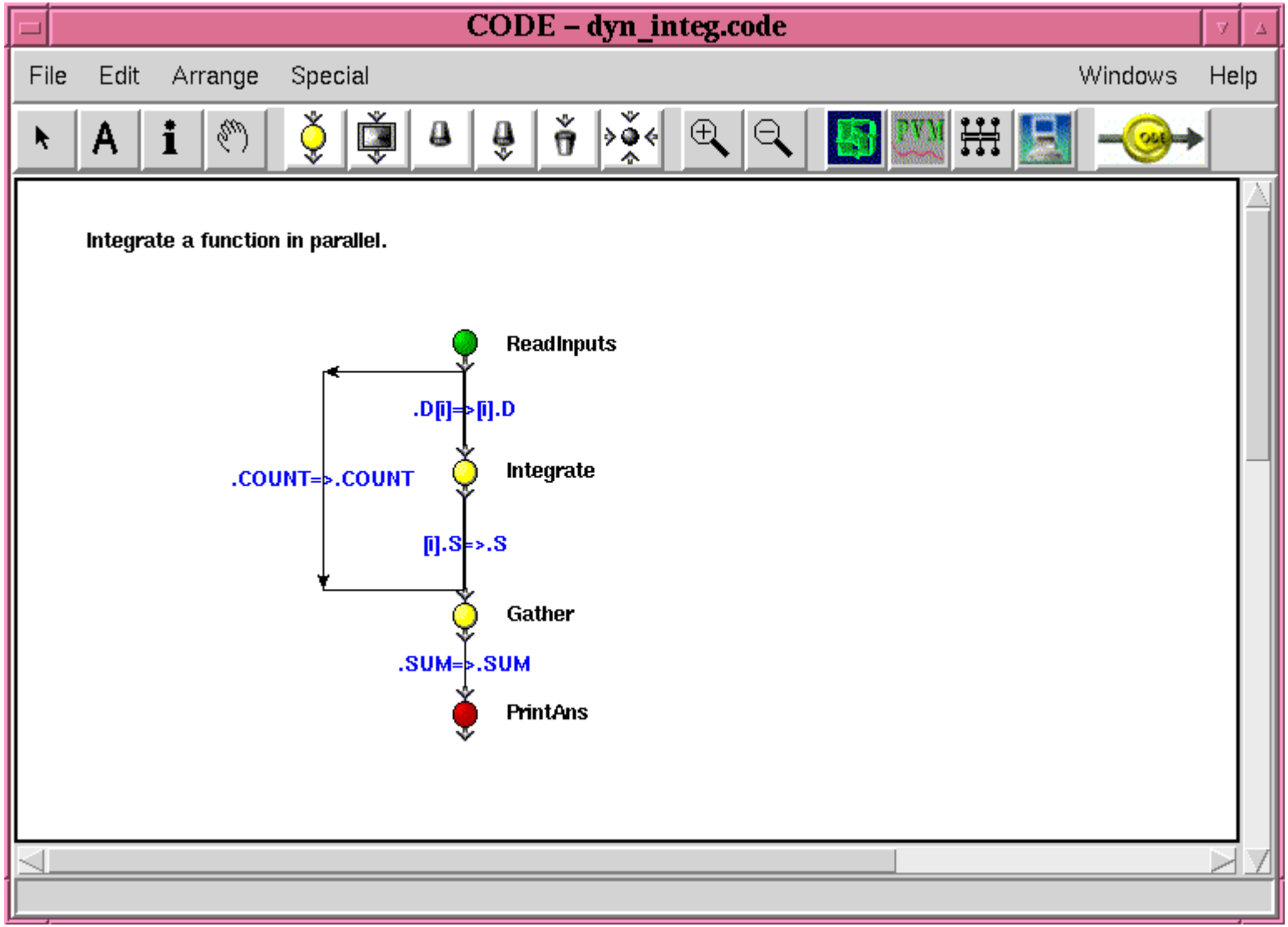
Недостатки:

1) В то время как данная модель программирования существенно упрощает задачу программиста, выявляя неявный параллелизм – она является менее выразительной и не предоставляет средств явного задания распределения ресурсов и управления.

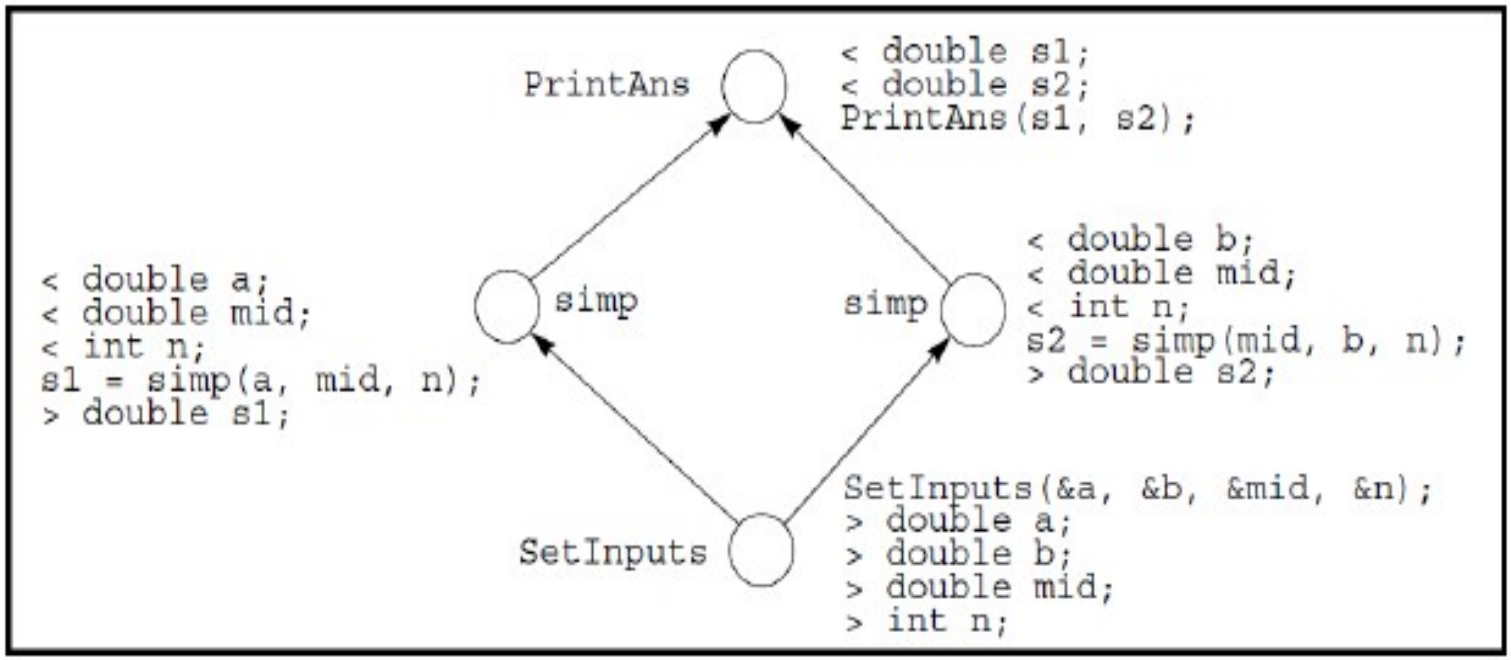
2) Представление программы в виде графа с узлами, которые соответствуют фрагментам вычислений, в отдельных случаях может привести к огромным и сложным графам.

3) Фрагменты данных в программе явно не выражены.

CODE


CODE – dyn_integ.code
File
Edit
Arrange
Special
Windows
Help
Integrate a function in parallel.
ReadInputs
.D[i]=>[i].D
.COUNT=>.COUNT
Integrate
[i].S=>.S
Gather
.SUM=>.SUM
PrintAns

HeNCE


PrintAns
< double s1;
< double s2;
PrintAns(s1, s2);
< double a;
< double mid;
< int n;
s1 = simp(a, mid, n);
> double s1;
simp
simp
< double b;
< double mid;
< int n;
s2 = simp(mid, b, n);
> double s2;
SetInputs
SetInputs(&a, &b, &mid, &n);
> double a;
> double b;
> double mid;
> int n;

VPE

VPE использует модель обмена сообщениями.

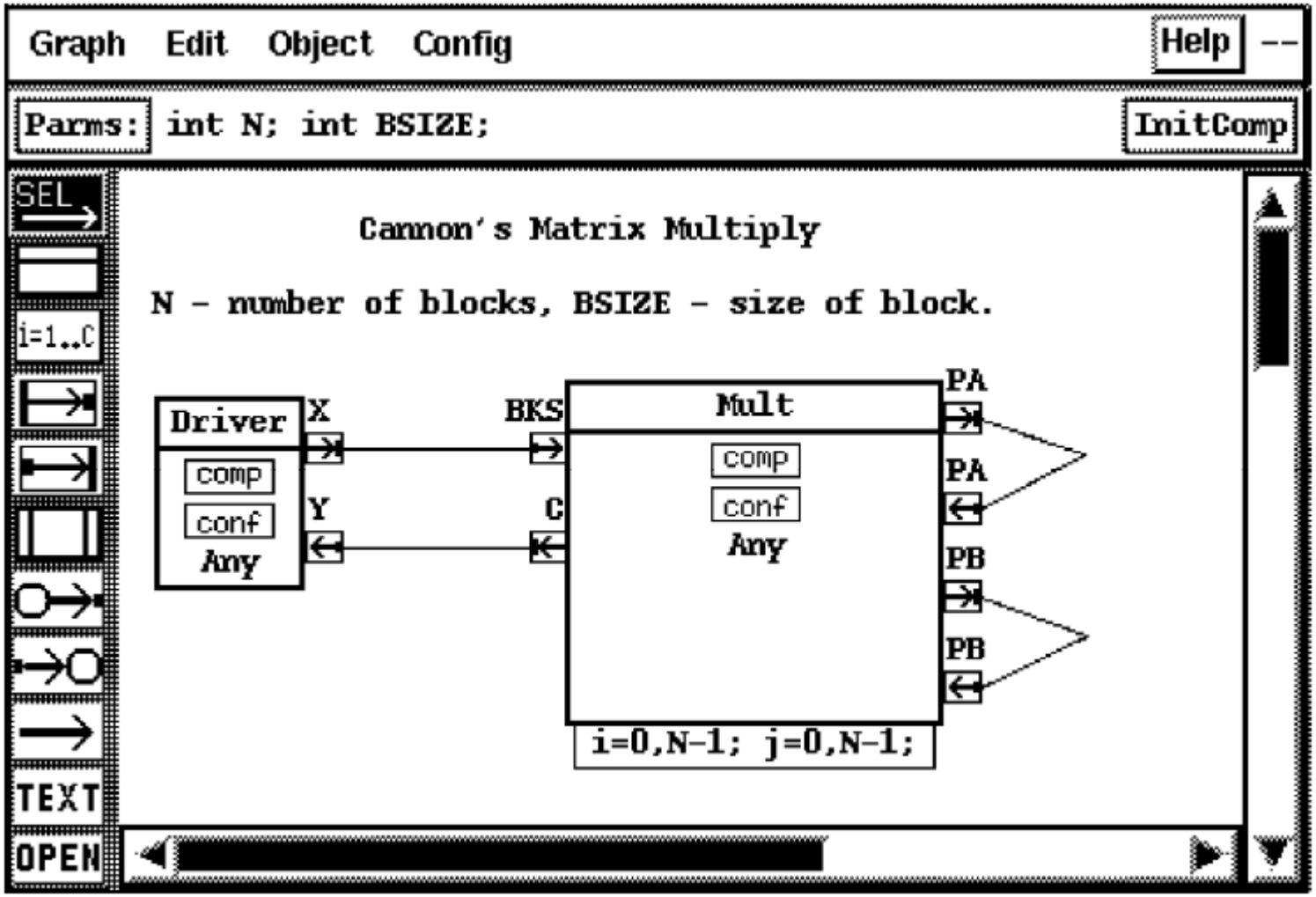
Преимущества:

1) Графические представления делают явный параллелизм понятнее.

Недостатки:

1) Программист должен следовать модели обмена сообщениями и явно указывать обмен данными между процессами в форме сообщений.

VPE


Graph
Edit
Object
Config
Help
--
Parms:
int N; int BSIZE;
InitComp
SEL
i=1..C
TEXT
OPEN
Cannon's Matrix Multiply
N - number of blocks, BSIZE - size of block.
Driver
comp
conf
Any
X
Y
BKS
C
Mult
comp
conf
Any
PA
PA
PB
PB
i=0,N-1; j=0,N-1;

Операторная схема

Операторная схема — это аналитическая форма представления алгоритма с помощью операторов, описывающих содержание некоторых автономных этапов вычислительного процесса, выделенных при решении задачи.

Операторная схемы была взята за основу в данной работе, так как:

1) она позволяет наглядно (графически) представлять алгоритм,

2) выполнять эквивалентные преобразования алгоритмов для достижения определенных свойств (решать задачу оптимального отображения алгоритма на вычислительную систему).

Базовая модель (Мальцева)

- Множество переменных x,y,z,... ∈ X (предметные символы)
- Множество функций f,g,h,... ∈ F (функциональные символы)
- Специальные символы – левая и правая скобки и запятая { “(“, “,”, “)” }
- Множество термов T:
  - Терм длины 1 – переменная
  - $f(a_1, \ldots, a_n)$ – терм, где $a_1, \ldots, a_n$ – термы меньшей длины

  Пример: f(x,y,g(z,x)) ∈ T
- Оператор суперпозиции: $g(x_1, \ldots, x_m) = f(f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m))$
- Оператор примитивной рекурсии: $f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$,

  $f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$
- Оператор минимизации: $\mu_y(f(x_1, \ldots, x_{n-1}, y) = x_n)$

Базовая модель (Мальцева)

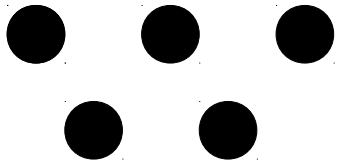
Частичная функция называется частично рекурсивной, если она может быть получена из простейших функций конечным числом операций подстановки, примитивной рекурсии и минимизации.

Тезис Черча:

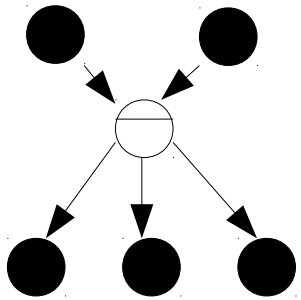
Класс алгоритмически (или машинно) вычислимых числовых функций совпадает с классом всех частично рекурсивыных функций.

Модель Visual LuNA++

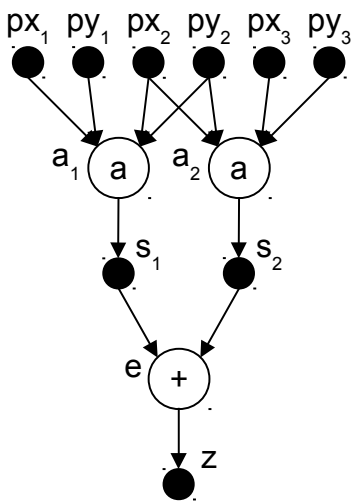
- Переменные (фрагменты данных)



- V-операции
  - предусловия

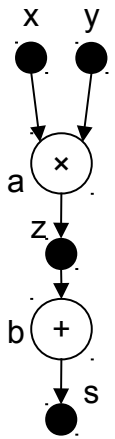


- V-алгоритмы

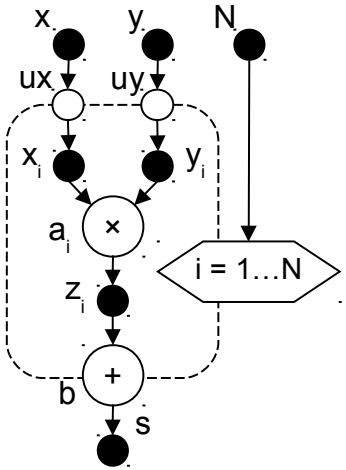


Модель Visual LuNA++

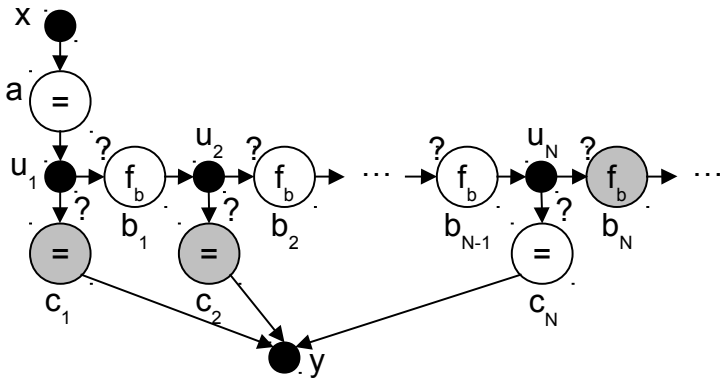
- Оператор суперпозиции (стыковки)



- Оператор for



- Оператор while



Модель Visual LuNA++

- Структурированные операции (оператор подстановки)
- V-алгоритмы можно свести к определению частично рекурсивных функций, следовательно они тоже задают частично рекурсивные (т.е. вычислимые) функции

Планы на следующий семестр

- Доработка модели языка визуального параллельного программирования для записи численных алгоритмов
- Разработка способов графического отображения алгоритмов на визуальном языке
- Доработка модели визуального языка и средств отображения для представления фрагментированных алгоритмов

Планы на второй год магистратуры

- Разработка и реализация системы визуального фрагментированного программирования

Спасибо за внимание