

Новосибирский государственный университет  
Факультет информационных технологий  
Кафедра параллельных вычислений

# **Алгоритмы редукции для системы фрагментированного программирования**

Подготовил  
Студент ФИТ НГУ  
А. Р. Сафин

Научный руководитель  
С. Е. Киреев

Новосибирск  
2013

# Технология фрагментированного программирования (ФП)

## Цель

Автоматизация реализации численных моделей.

## Суть

Фрагментация.

Отделение алгоритма от реализационной части.

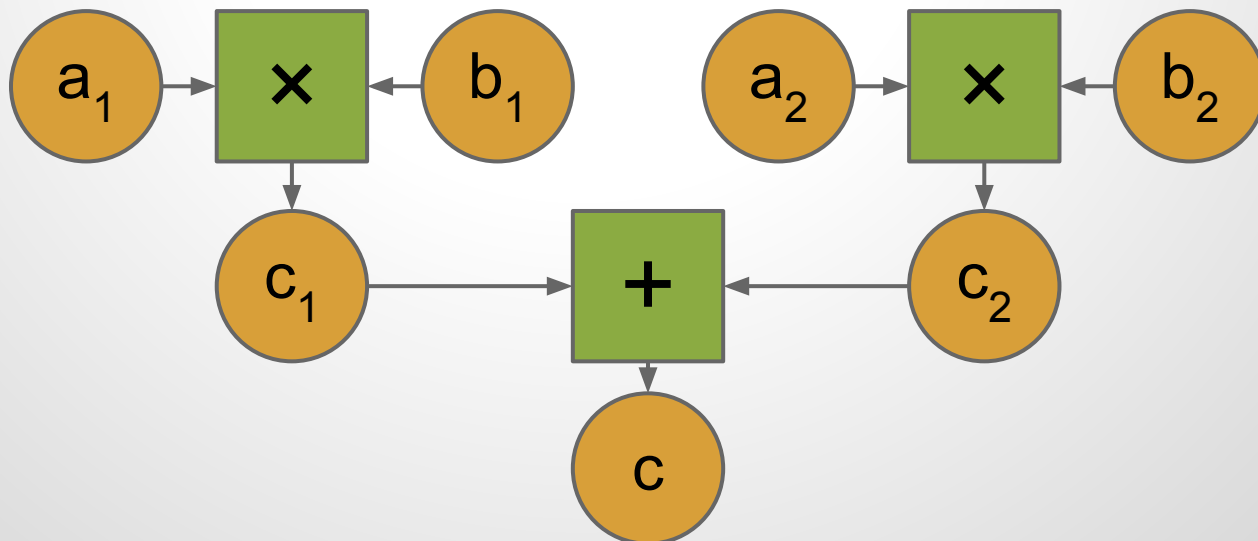
Исполнительная система.

# Технология ФП

Алгоритм состоит из

- фрагментов данных,
- фрагментов вычислений,
- отношений между ними.

Пример алгоритма:  $c = a_1 \times b_1 + a_2 \times b_2$



# Редукция

## Полная редукция

Результат - одно значение.

## Частичная редукция

Результат - последовательность  
частичных результатов.

# Операция полной редукции

Пусть у нас есть:

- Множество  $S$
- Ассоциативный коммутативный оператор

$$\circ : S \times S \rightarrow S$$

- Множество переменных  $A$

$$A = \{a_1, a_2, \dots, a_n\}; a_i \in S$$

Полная редукция определяется как

$$\text{reduce}(A, \circ) = a_1 \circ a_2 \circ \dots \circ a_n$$

# Операция частичной редукции

Пусть у нас есть:

- Счетное множество  $V$

$$V = \{b_1, b_2, \dots\}; b_i \in S$$

- Счетная последовательность  $V_s$

$$V_s = \{V_1, V_2, \dots\}$$

$$V_i \subseteq V_{i+1}; V_i \subseteq V$$

Частичная редукция определяется как

$$\text{reduce}_p(V_s, \circ) = \{\text{reduce}(V_1, \circ), \text{reduce}(V_2, \circ), \dots\}$$

# Особенности алгоритма редукции в системе ФП

- Является фрагментом вычислений.
- Должен учитывать конфигурацию вычислителя.
- Должен поддерживать произвольные операторы и типы данных.
- Исполняется на вовлечённых узлах вычислителя.
- Входные фрагменты данных поступают на вовлечённые узлы, порядок поступления не определён.
- Обеспечивает сбор результата на целевом узле.

# Оценка алгоритма редукции

## Полная редукция

Общее время работы

## Частичная редукция

Время запаздывания, т.е. интервал времени между получением входного фрагмента и учётом его в очередном частичном результате.



# Постановка задачи

## Цель работы

Разработка алгоритма асинхронной распределённой редукции для использования в системе фрагментированного программирования.

# Обзор существующих алгоритмов редукции

## Open MPI

Полная редукция

Биномиальное дерево

## MPICH

Полная редукция

Редукция векторов

## Charm++

Полная редукция

Дерево заданной степени

# Идеи алгоритма

## Дерево редукции

Листья - входные переменные.

Остальные вершины содержат значение редукции своих поддеревьев

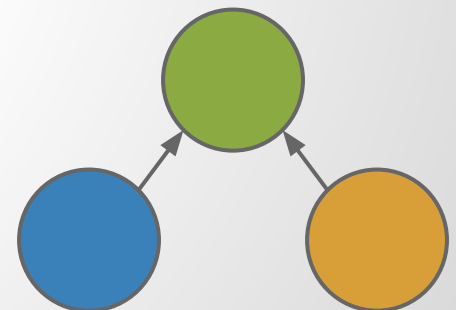
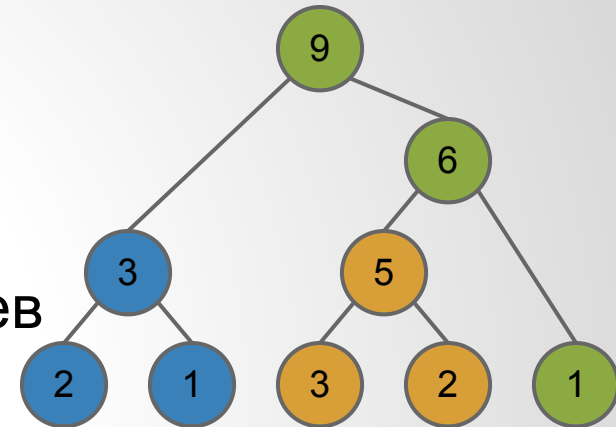
## Дерево вычислителя

Вершины - узлы вычислителя.

Дуги - возможность передачи значений фрагментов

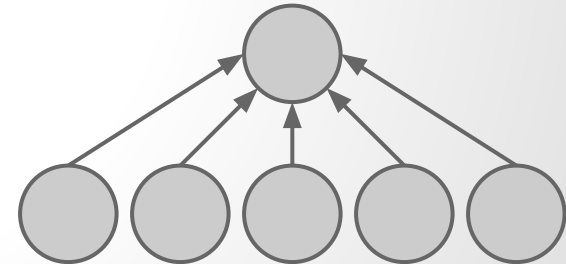
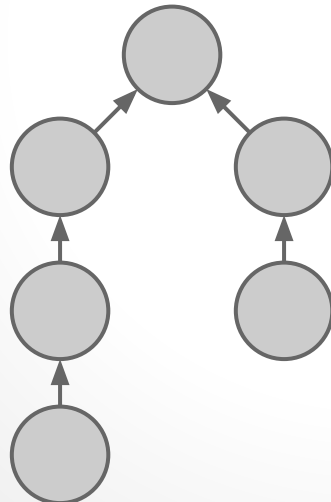
## Алгоритм работы узла

Описывает редукцию фрагментов в пределах одного узла.



# Построение дерева вычислителя

- Децентрализованный алгоритм построения дерева
- Баланс между высотой и степенью дерева



$P$  - степень дерева, один из параметров алгоритма редукции.

# Алгоритм работы узла

## Задачи узла

- Принимать входные фрагменты данных.
- Принимать фрагменты данных от потомков в дереве.
- Выполнять редукцию принимаемых фрагментов данных.
- Отправлять редуцированные фрагменты предку, либо системе исполнения.

## Сигнал завершения

- Полная редукция: система сообщает об окончании поступления входных фрагментов.
- Частичная редукция: система посылает сигнал завершения

# Алгоритм работы узла

## Полная редукция

Результат отправляется только один раз, перед завершением работы.

## Частичная редукция

Результаты отправляются с периодом

$T_{wait}$ .

$T_{wait}$  - параметр алгоритма.

# Эмулятор работы алгоритма

- Дискретное модельное время
- Модель вычислительной системы
  - N узлов
  - Связь «каждый с каждым»
- Генерация входных фрагментов
- Эмуляция работы алгоритмов полной и частичной редукции

# Параметры эмулятора работы алгоритма

## Параметры вычислительной системы

$N$  - число узлов,

$T_{lat}$  - латентность сети,

$T_{bth}$  - время передачи одного фрагмента,

$T_{op}$  - время редуцирования двух операндов.

## Параметры задачи

$N_f$  - количество генерируемых фрагментов на узел,

$T_{Gmin}$ ,  $T_{Gmax}$  - параметры темпа генерации фрагментов.

## Параметры алгоритма

$P$  - степень дерева вычислителя,

$T_{wait}$  - период времени между отправками.



# Результат работы эмулятора

## Полная редукция

$T_{res}$  - общее время работы.

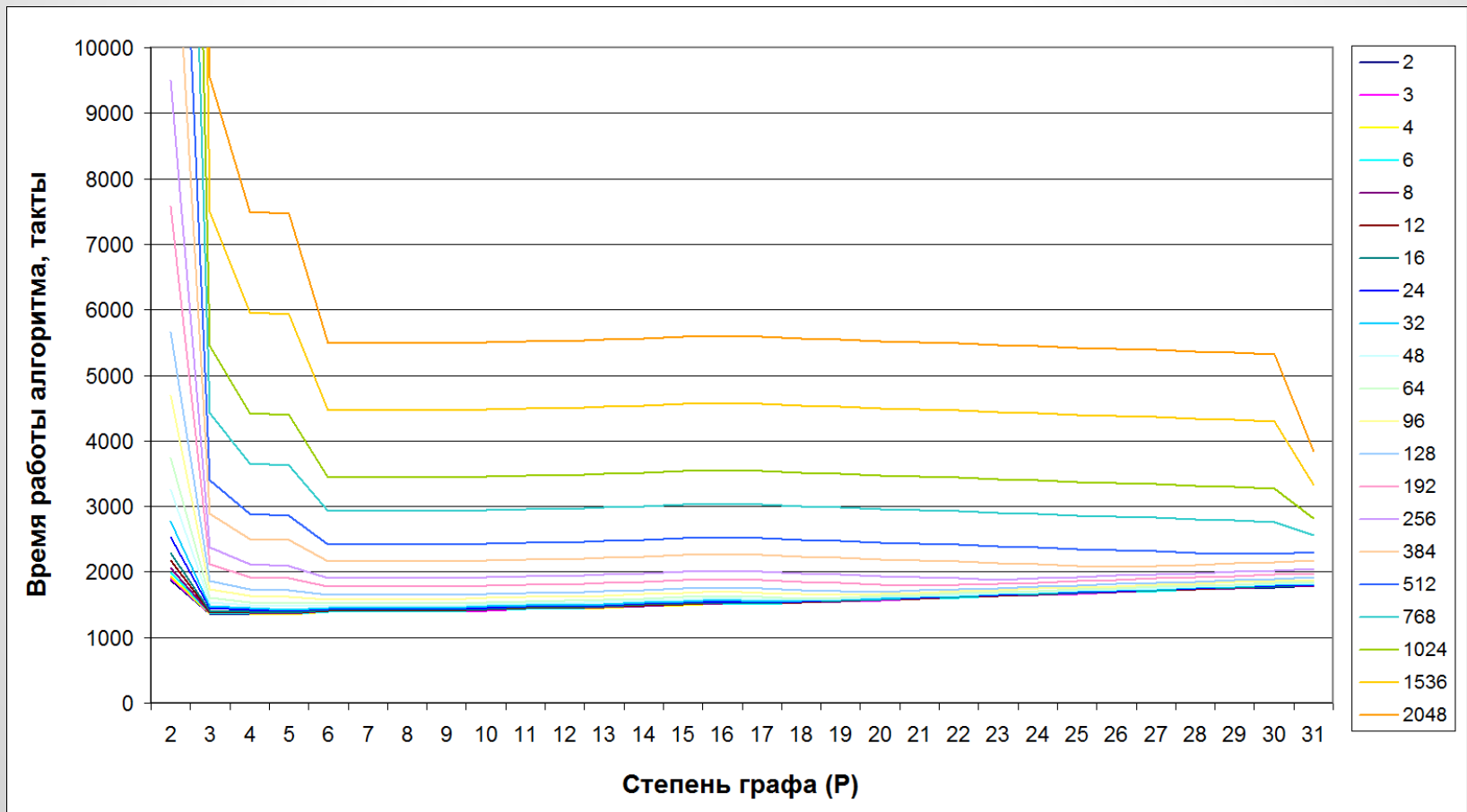
## Частичная редукция

Зависимость  $\alpha_n$  от времени,

где  $\alpha_n$  - среднее время запаздывания  
последних  $n$  учтённых фрагментов.

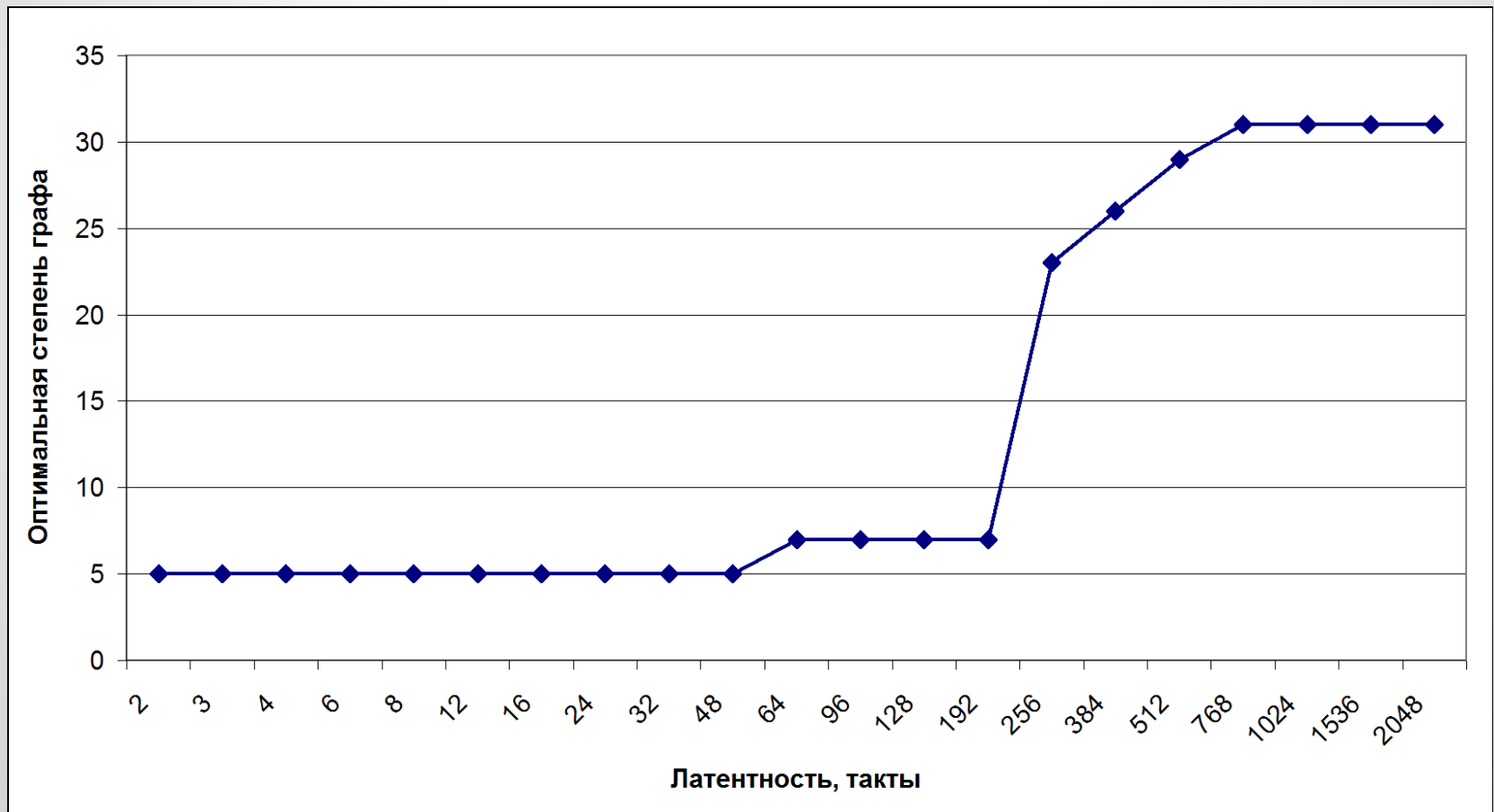
# Результаты тестирования алгоритма полной редукции

Зависимость времени работы алгоритма от степени дерева вычислителя для разных величин латентности



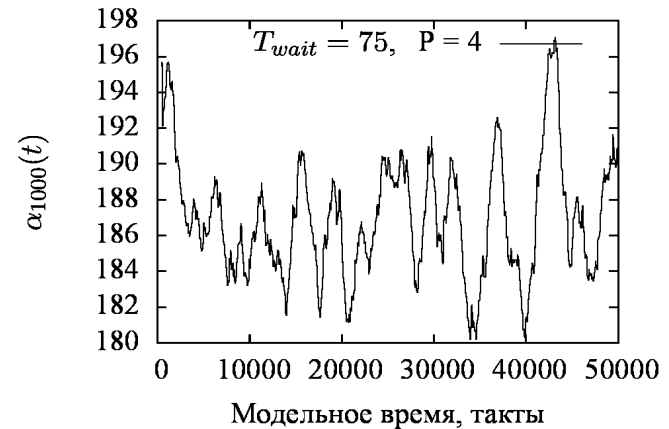
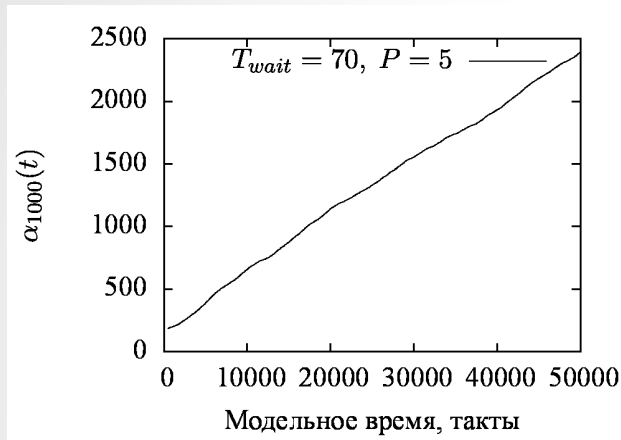
# Результаты тестирования алгоритма полной редукции

Зависимость оптимальной степени дерева вычислителя от латентности



# Результаты тестирования алгоритма частичной редукции

Динамика времени запаздывания для различных параметров алгоритма



Среднее время запаздывания для различных параметров алгоритма

		$T_{wait}$																							
		10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125
$P$	1	689	691	692	693	698	701	711	720	733	739	738	740	770	787	818	833	865	865	920	909	960	945	1023	1054
	2	∞	∞	∞	∞	179	182	186	192	197	203	209	213	218	226	231	236	247	254	254	270	276	283	291	301
	3	∞	∞	∞	∞	∞	∞	∞	173	174	177	181	186	191	195	200	206	213	217	227	230	236	245	250	253
	4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	190	180	183	189	193	199	205	210	220	223	228	236	243	247
	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	184	187	190	195	198	207	211	218	223	228
	6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	200	203	205	209	212	219	228
	7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	228	221	222	229
	8	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	257

# Заключение

- Выполнен обзор ряда существующих алгоритмов редукции.
- Разработаны параметризованные алгоритмы для полной и частичной редукции.
- Разработан эмулятор, позволяющий найти оптимальные значения параметров алгоритма при заданных параметрах вычислителя.

## Дальнейшие планы:

- Определение оптимальных параметров алгоритма для реальных вычислителей.
- Расширение алгоритма для учёта динамической балансировки нагрузки.
- Разработка алгоритма редукции для суммирования с заданной точностью.

# Работа была представлена на МНСК

Сафин А. Р. Разработка и реализация алгоритма распределенной асинхронной редукции для системы фрагментированного программирования // Материалы 51-ой юбилейной международной научной студенческой конференции «Студент и научно-технический прогресс»: Информационные технологии / Новосиб. гос. ун-т. Новосибирск, 2013. - с. 214.

**Спасибо  
за внимание.**