

Использование компиляторов языка Фортран при решении вычислительных задач

Евгений Алексеев, Денис Лутошкин,
Александр Огородов, Вячеслав Стародумов
Вятский Государственный Университет
ealekseev@gmail.com

Умножение матриц

		Классический алгоритм, порядок выполнения циклов						Блочный алгоритм (размер блока 16)	matmul
		ijk	jik	kij	ikj	jki	kji		
1536	gfortran	20.5	23.65	52.9	49.9	2.22	3.77	2	0.5
	ifort	0.67	0.7	0.7	0.67	1.99	0.7	2.68	0.69
	pgfortran	1.86	2.05	2.05	1.9	1.9	2.08	5.64	0.91
2048	gfortran	91.12	89.84	219.5	222.48	5.26	8.84	4.75	1.16
	ifort	1.52	1.66	1.65	1.6	4.68	1.64	6.98	1.63
	pgfortran	4.46	4.86	4.91	4.54	4.54	4.51	61.78	2.19

Быстродействие программ умножения матриц

- Проприетарные компиляторы оптимизируют исходный код лучше, чем gfortran
- Программы на Фортране работают на 10-12% быстрее чем программы на С
- Для решения задач, сводящихся к умножению матриц следует использовать matmul и компилятор gfortran-7

Решение СЛАУ методом Гаусса

Размерность системы	Компилятор Фортрана	Время счёта, с	Компилятор C(C++)	Время счёта, с
1000	gfortran	3.78	g++	0.89
	ifort	3.9	icpc	0.88
	pgfortran	1.44	pgc++	0.85
2000	gfortran	34.37	g++	10.10
	ifort	33.02	icpc	10.11
	pgfortran	16.13	pgc++	9.95

Решение СЛАУ

итерационными методами

Размерность системы	Компилятор	Метод Зейделя			Метод простой итерации		
		k	1	2	k	1	2
10000	gfortran	12	3.58	3.58	37	10.4	2.32
	pgfortran		3.48	3.48		2.28	2.27
	ifort		3.5	3.5		7.5	2.19
12000	gfortran	13	6.31	6.31	37	18.2	3.34
	pgfortran		5.6	5.6		3.28	3.25
	ifort		5.64	5.64		12.8	3.17

Выводы

- Значительное ускорение наблюдается, если получается написать код с использованием только матричных подпрограмм и операторов
- Коммерческий проприетарный компилятор pgfortran генерирует быстродействующее параллельное приложение при написании программы без использования современных матричных операторов и подпрограмм

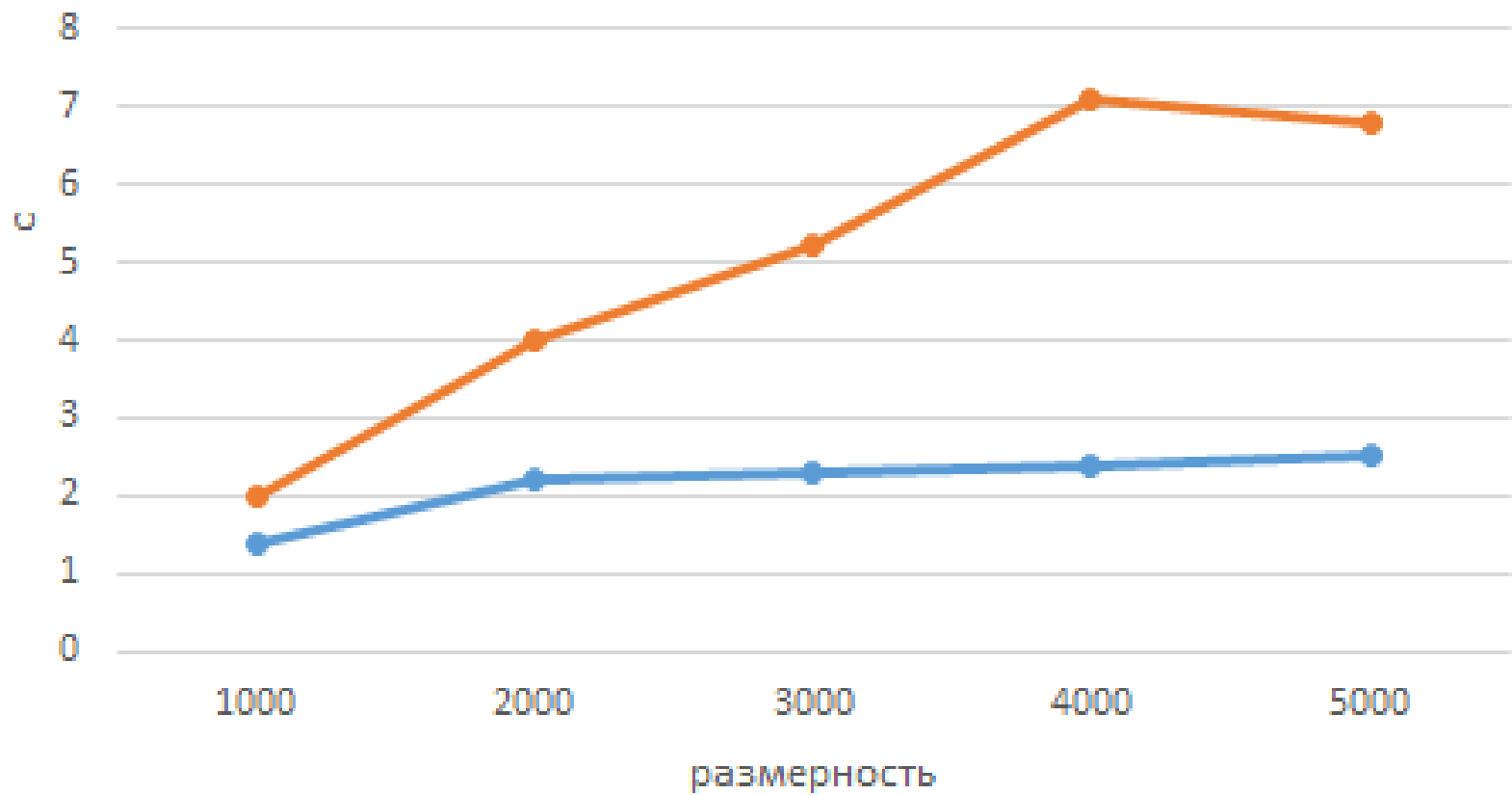
Сравнение итерационных алгоритмов

Размерность системы	Компилятор	Метод Зейделя		Метод простой итерации
		обычный	усовершенствованный	
10000	gfortran	3.57	2.61	2.32
	pgfortran	3.48	2.54	2.27
	ifort	3.5	2.48	2.19
12000	gfortran	6.26	4.01	3.34
	pgfortran	5.6	3.78	3.25
	ifort	5.62	3.76	3.17

Ускорение программ умножения матриц

(автораспараллеливание, 4 ядра)

размерность	matmul	Классический алгоритм	Блочный алгоритм
1000	1.4	1.4	2
2000	2.2	2.2	4
3000	2.3	2.3	5.2
4000	2.4	2.4	7.1
5000	2.5	2.5	6.8

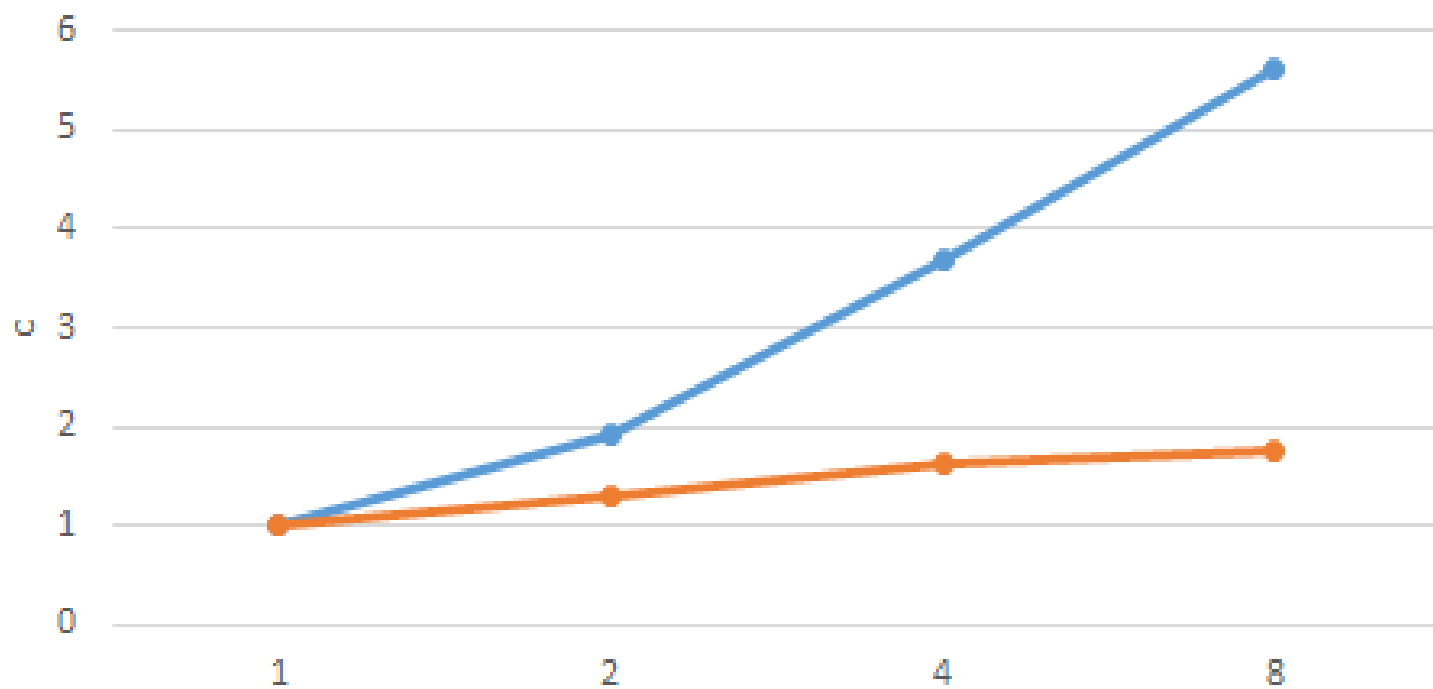


—●— Классический алгоритм и matmul

—●— Блочный алгоритм

Ускорение при решении СЛАУ с использованием технологии КОМАССИВОВ

	Количество процессорных узлов			
	1	2	4	8
Метод простой итерации	1	1.93	3.68	5.6
Метод Зейделя	1	1.3	1.64	1.75



Количество процессорных узлов

—●— Метод простой итерации —●— Метод Зейделя

Спасибо за внимание!
Готов ответить на
ваши вопросы.