



Russian Supercomputing Days
Суперкомпьютерные дни в России
24 сентября 2018 г.

**Параллельный
предобуславливатель на
основе степенного
разложения обратной
матрицы для решения
разреженных линейных
систем на графических
процессорах**

Н.В. Репин,
А.В. Юлдашев (art@ugatu.su)

Уфимский государственный авиационный
технический университет

Введение

Решение систем линейных алгебраических уравнений (СЛАУ) с разреженной матрицей является вычислительным ядром множества задач математического моделирования, в том числе задачи моделирования фильтрационных течений углеводородов в пористой среде. Время решения СЛАУ при численном решении задачи фильтрации углеводородов может достигать от 60 до 90 % от всего времени расчета.

Цель: повышение масштабируемости двухступенчатого предобуславливателя CPR на гибридных системах с несколькими графическими процессорами NVIDIA.

Задачи:

- разработка параллельного предобуславливателя на основе степенного разложения обратной матрицы (в качестве альтернативы классическому AMG);
- разработка параллельного алгоритма решения на графических процессорах NVIDIA трехдиагональных систем специального вида;
- оценка масштабируемости параллельного предобуславливателя на гибридной системе с несколькими GPU NVIDIA.

Системы алгебраических уравнений

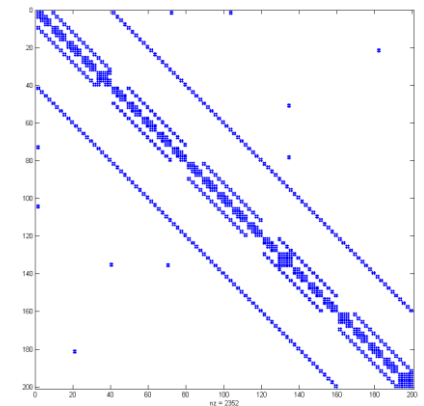
Для решения систем нелинейных алгебраических уравнений после дискретизации уравнений многофазной фильтрации с помощью метода конечных объемов по пространству и полностью неявной схемы по времени используется метод Ньютона:

$$J(x^n) \delta x^n = -F(x^n), \quad x^{n+1} = x^n + \delta x^n$$

В случае двухфазной фильтрации разреженная матрица Якоби состоит из блоков:

$$J_{ij}(x^n) = \begin{bmatrix} \frac{\partial R_w^i}{\partial S_w^j} & \frac{\partial R_w^i}{\partial p^j} \\ \frac{\partial R_o^i}{\partial S_w^j} & \frac{\partial R_o^i}{\partial p^j} \end{bmatrix}$$

R_w – левая часть дискретного уравнения для водной фазы.
 R_o – левая часть дискретного уравнения для нефтяной фазы.
 i, j – индексы ячеек сетки дискретизации.



■ - блок размера $k \times k$, где k - количество фаз (1-3)

Особенности: размерность 10^6 - 10^7 , заполненность менее 0,01 %, плохая обусловленность, несимметричность, блочная структура, наличие регулярной и нерегулярной частей матрицы.

Используемые методы решения систем линейных алгебраических уравнений

Используемый итерационный метод решения:

Метод бисопряженных градиентов со стабилизацией (BiCGStab)

Используемые предобуславливатели:

- двухступенчатый CPR
 - классический алгоритм AMG (алгебраический многосеточный метод)
 - аппроксимация обратной матрицы на основе степенного разложения – Approximation of Inverse by Power Series, AIPS
 - блочная модификация неполного LU-разложения без заполнения BILU(0)

Алгоритм решения СЛАУ с

предобуславливателем CPR $M_{CPR}x = b$:

1. $r_p = b_p - D_{ps}D_{ss}^{-1}b_s;$

2. $A_p = A_{pp} - D_{ps}D_{ss}^{-1}A_{sp};$

3. $A_p\delta x_p = r_p;$

4. $\delta x = (\delta x_p, 0)^T;$

5. $b = b - A\delta x;$

6. $M_{ILU}x = b;$

7. $x = x + \delta x.$

Использование метода AIPS в качестве первой ступени CPR

A_p - матрица решаемой СЛАУ на давление

$$A_p = P + R$$

$$A_p = P(E - (-1)P^{-1}R)$$

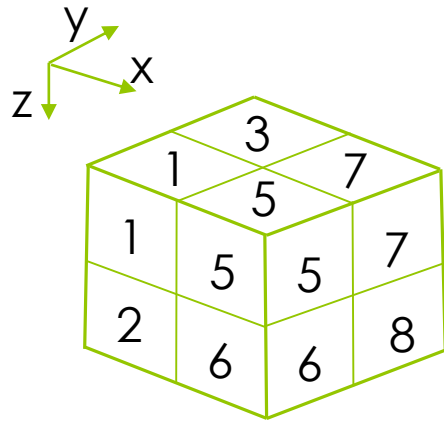
$$A_p^{-1} \approx M_N^{-1} = \left[E + \sum_{k=1}^N (-1)^k (P^{-1}R)^k \right] P^{-1}$$

В качестве P берется трехдиагональная часть матрицы A_p .

$$A_p \delta x_p = r_p \quad \longrightarrow \quad \delta x_p \approx M_N^{-1} r_p = P^{-1} r_p + \sum_{k=1}^N (-1)^k (P^{-1}R)^k P^{-1} r_p$$

- 1) Умножение разреженной матрицы R на вектор
- 2) Решение СЛАУ с трехдиагональной матрицей P

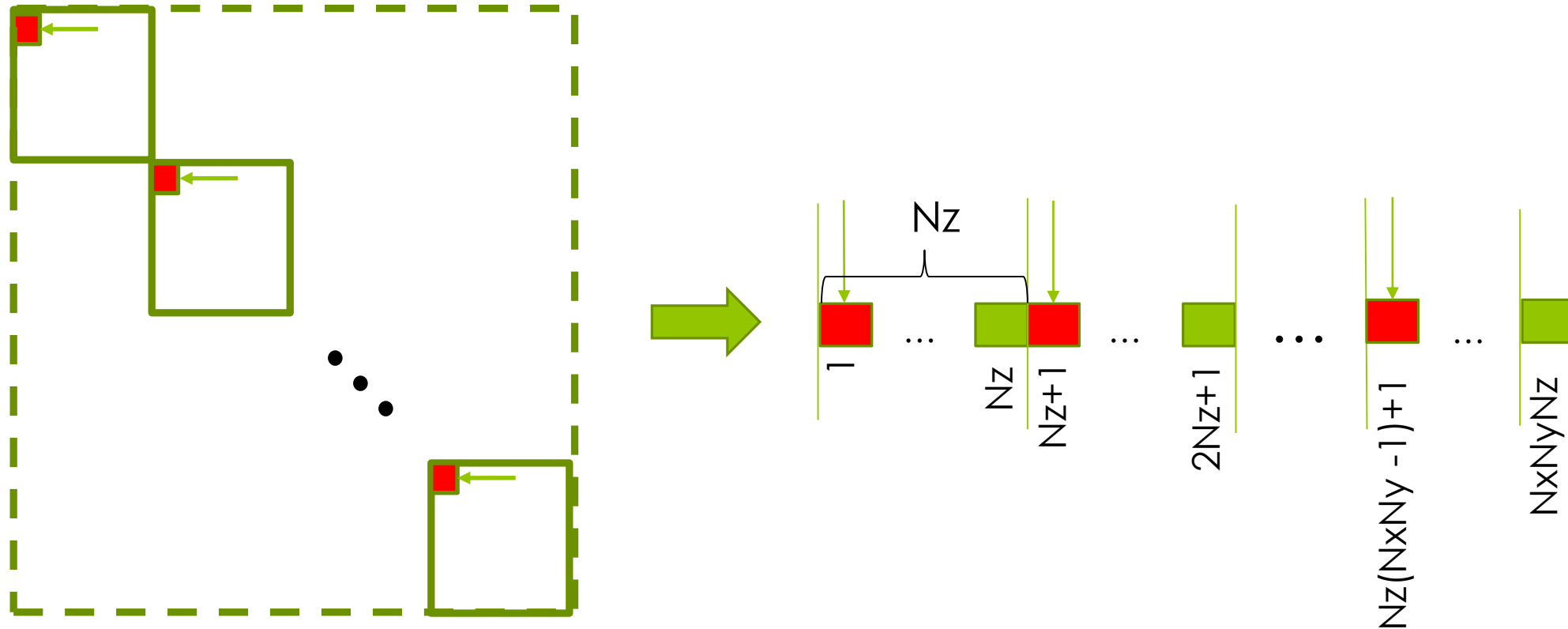
Трёхдиагональная часть матрицы на давление (блоки одинакового размера)



d_1	u_1	x	x				
l_2	d_2		x	x			
x		d_3	u_3			x	
	x	l_4	d_4				x
x				d_5	u_5	x	
	x			l_6	d_6		x
		x	x			d_7	u_7
			x	x		l_8	d_8

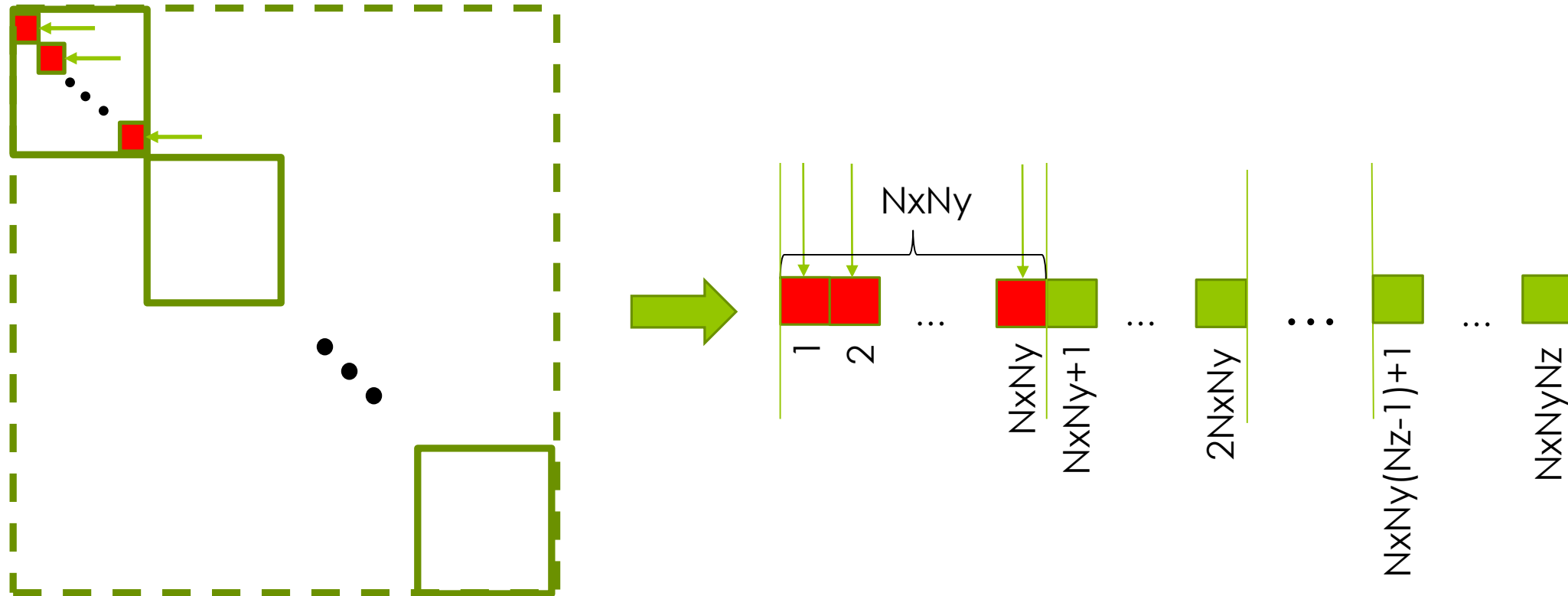
d – элементы главной диагонали
 u – элементы верхней диагонали
 l – элементы нижней диагонали
 x – элементы, вне трёхдиагональной части

Оптимизация работы с памятью на GPU



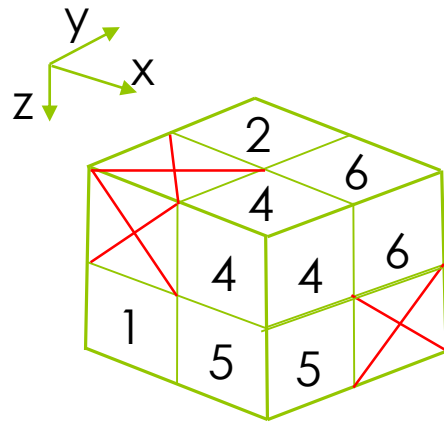
Для достижения лучшей производительности элементы трехдиагональной матрицы переставляются таким образом, чтобы обращение шло к последовательно расположенным адресам в памяти.

Оптимизация работы с памятью на GPU



Для достижения лучшей производительности элементы трехдиагональной матрицы переставляются таким образом, чтобы обращение шло к последовательно расположенным адресам в памяти.

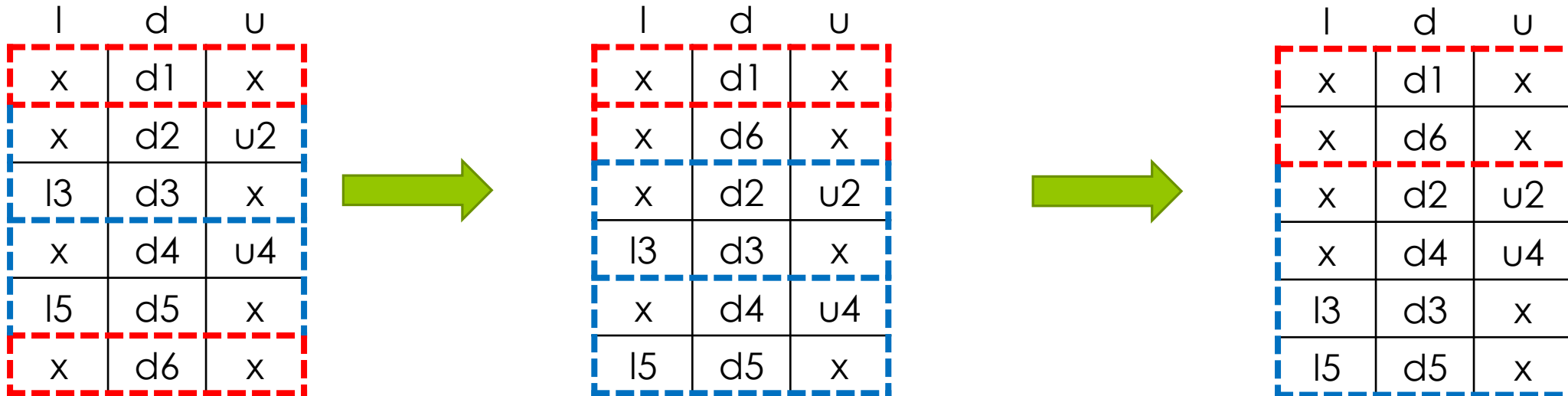
Трехдиагональная часть матрицы на давление (блоки разного размера)



d_1	x	x			
	d_2 u_2				x
x	l_3 d_3				
		d_4 u_4	x		
x		l_5 d_5			
	x	x			d_6

d – элементы главной диагонали
 u – элементы верхней диагонали
 l – элементы нижней диагонали
 x – элементы, вне трехдиагональной части

Балансировка нагрузки



Выделяем блоки одинаковой размерности. Переставляем элементы таким образом, чтобы блоки выстроились согласно их размерности по возрастанию. Причем каждая последовательность блоков одинаковой размерности может асинхронно обрабатываться отдельным вычислительным ядром. К каждой последовательности блоков применяется предыдущая оптимизация.

Алгоритм решения СЛАУ с трехдиагональной матрицей

1. *Этап анализа структуры матрицы.*
 1. Провести анализ структуры трехдиагональной матрицы, результатом которого является выявление независимых блоков, их размеров и местоположения.
2. *Этап подготовки матрицы.*
 1. Отсортировать блоки матрицы согласно их размерности и выделить группы блоков одинаковой размерности в целях балансировки нагрузки на этапе решения.
 2. Провести перестановку строк матрицы внутри каждой группы блоков одинаковой размерности для более производительного доступа к памяти на этапе решения.
3. *Этап решения трехдиагональной линейной системы.*
 1. Провести перестановку элементов вектора правой части согласно перестановкам строк матрицы, проведенным на шагах 2.1 и 2.2.
 2. Параллельно применить метод прогонки для каждого блока матрицы.
 3. Провести обратную перестановку элементов вектора решения.

Экспериментальная часть

Название матрицы СЛАУ	Размерность матрицы	Количество ненулевых элементов	Ср. кол-во ненулевых элементов в строке
imsh	1 500 000	55 815 624	37,210
immn	2 304 102	42 859 314	18,601
krrv	4 320 921	85 471 137	19,781
mmnt	5 637 747	109 595 799	19,440
fdrv	6 610 263	118 221 633	17,885
kmms	8 630 895	167 332 329	19,388
lkms	13 665 705	254 102 823	18,594

Название матрицы	Размерность матрицы	Мин. размер блока	Макс. размер блока	Кол-во независимых блоков
imsh3	500 000	10	10	500 00
immn3	768 034	1	34	221 402
krrv3	1 440 307	1	39	233 866
mmnt3	1 879 429	1	45	281 352
fdrv3	2 203 421	1	55	633 202
kmms3	2 876 965	1	121	389 199
lkms3	4 555 235	1	27	1 167 013

СЛАУ решаются методом BiCGStab с предобуславливателями CPR-AMG и CPR-AIPS.

Критерий остановки итерационного процесса: Относительная невязка $\frac{\|b - Ax^s\|}{\|b\|} \leq 10^{-4}$.

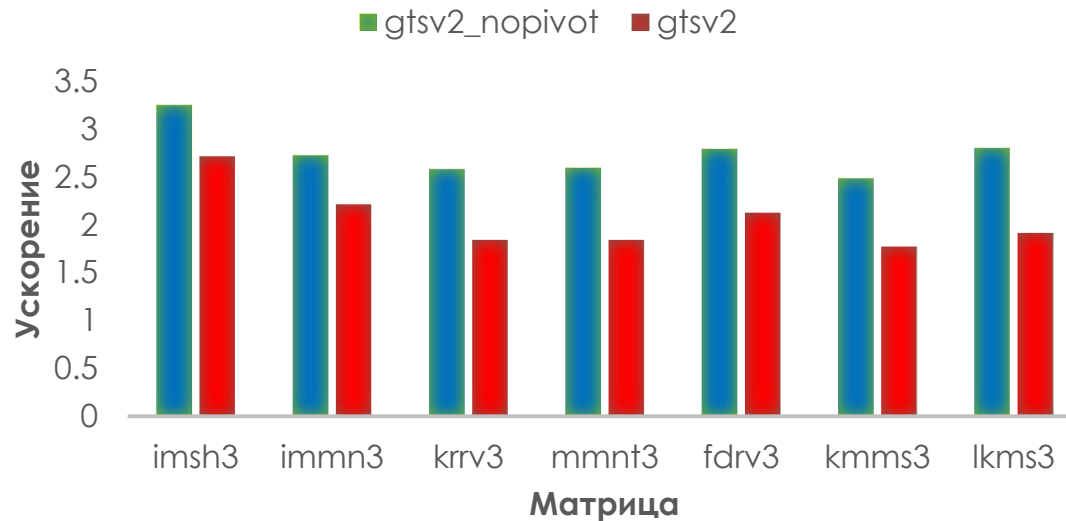
Вычислительные платформы с CUDA 9.1, AmgX v.2.0.0.130-opensource, OpenMPI v.1.8.8:

1) узел кластера УГАТУ со следующими характеристиками: 24-ядерная система 2 x Intel Gold 6126, 2 x GPU Nvidia P100 (3584 CUDA ядер), 192 Gb ОЗУ

2) платформа Amazon AWS p3.8xlarge 32 x vCPU Intel Xeon E5-2686 v4 и 4 x GPU NVIDIA Tesla V100 (5120 CUDA ядер)

Время решения СЛАУ с трехдиагональной матрицей

P100	Алгоритм	Матрица	imsh3	immn3	krrv3	mmnt3	fdrv3	kmms3	lkms3
	gtsv2	Время решения, мс	0,323	0,469	0,753	0,976	1,209	1,459	2,207
gtsv2_nopivot	0,386		0,578	1,055	1,374	1,586	2,046	3,228	
Предложенный алгоритм	0,119		0,212	0,409	0,530	0,568	0,824	1,152	
		Время подготовки, мс	0,130	0,190	0,400	0,550	0,710	0,960	1,760



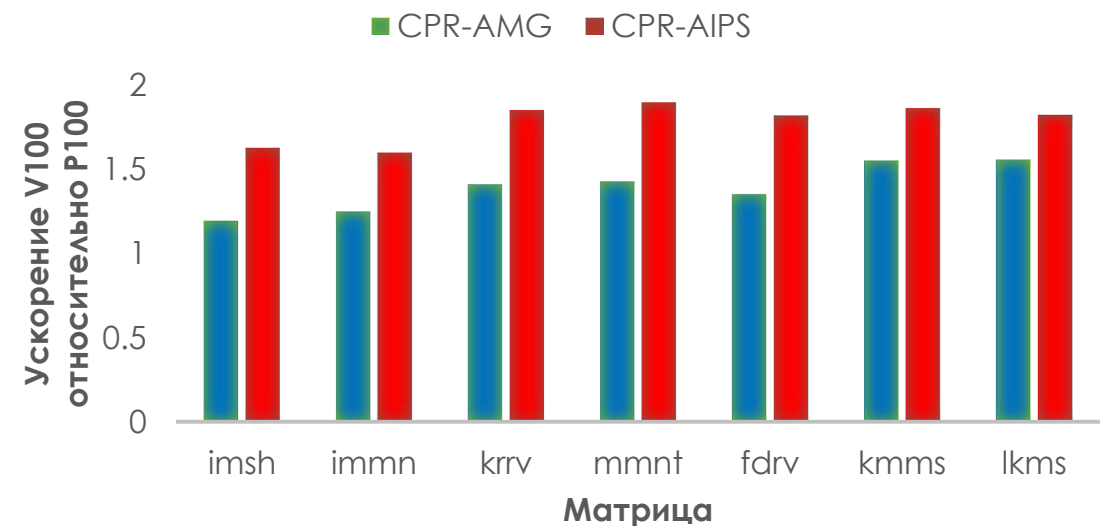
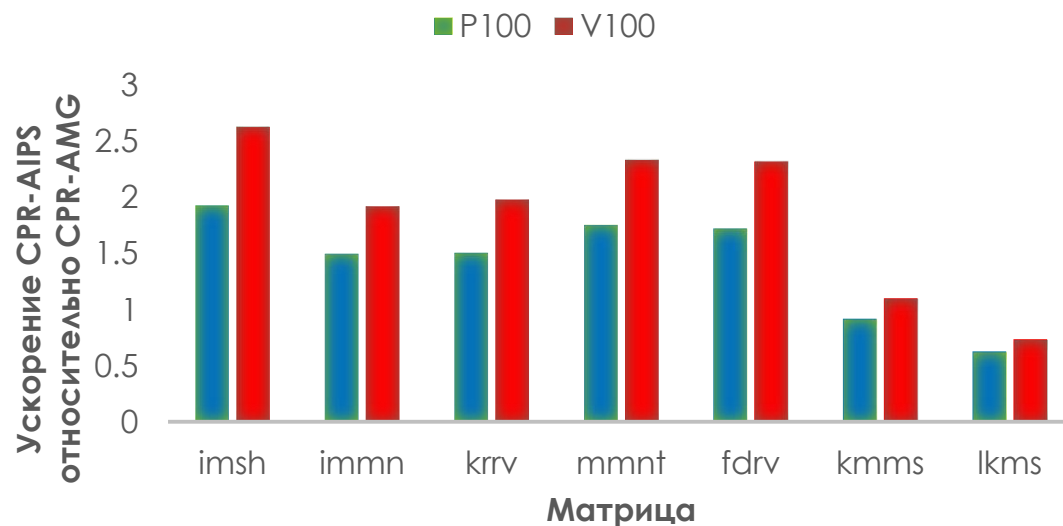
~ускорение в 2 раза
 относительно gtsv2
 ~ускорение в 2.75 раза
 относительно gtsv2_nopivot

gtsv2 — алгоритм SPIKE из библиотеки cuSPARSE

gtsv2_nopivot — комбинация методов циклической и параллельной циклической редукции из библиотеки cuSPARSE

Быстродействие CPR-AIPS

P100	Метод	Матрица	imsh	immn	krrv	mmnt	fdrv	kmms	lkms
	CPR-AMG	Время решения, с		0,135	0,18	0,285	0,36	0,314	0,678
Число итераций			1	2,5	2	2	1	3,5	2,5
CPR-AIPS	Время решения, с		0,07	0,12	0,189	0,205	0,182	0,738	1,409
	Число итераций		1	2,5	2	1,5	1	5	6,5
V100	Метод	Матрица	imsh	immn	krrv	mmnt	fdrv	kmms	lkms
	CPR-AMG	Время решения, с		0,113	0,144	0,202	0,252	0,232	0,437
Число итераций			1	2,5	2	2	1	3,5	2,5
CPR-AIPS	Время решения, с		0,043	0,075	0,102	0,108	0,1	0,396	0,772
	Число итераций		1	2,5	2	1,5	1	5	6,5

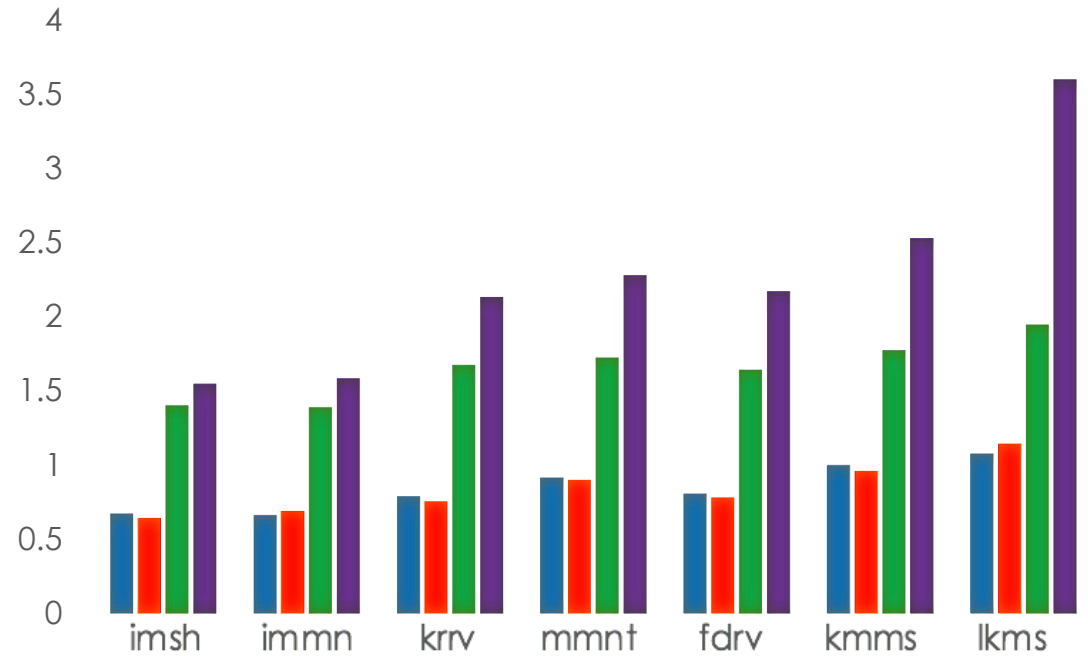


Масштабируемость CPR-AIPS

Матрица	Количество GPU	CPR-AMG		CPR-AIPS	
		1	4	1	4
imsh	Время решения, с	0,113	0,179	0,043	0,028
	Число итераций	1	1	1	1
immn	Время решения, с	0,144	0,230	0,075	0,047
	Число итераций	2,5	2,5	2,5	2,5
krrv	Время решения, с	0,202	0,298	0,102	0,048
	Число итераций	2	2	2	2
mmnt	Время решения, с	0,252	0,328	0,108	0,048
	Число итераций	2	2	1,5	1,5
fdrv	Время решения, с	0,232	0,303	0,100	0,046
	Число итераций	1	1	1	1
kmms	Время решения, с	0,437	0,514	0,396	0,157
	Число итераций	3,5	5	5	4,5
lkms	Время решения, с	0,568	0,590	0,772	0,215
	Число итераций	2,5	4	6,5	6,5

УСКОРЕНИЕ НА 2-4 GPU ОТНОСИТЕЛЬНО 1 GPU

■ CPR-AMG 2 V100 ■ CPR-AMG 4 V100
 ■ CPR-AIPS 2 V100 ■ CPR-AIPS 4 V100



- ~ускорение CPR-AIPS при использовании 2 V100 1.64 раза
- ~ускорение CPR-AIPS при использовании 4 V100 2.26 раза
- ускорение CPR-AIPS относительно CPR-AMG от 2.6 до 5.2 раза

Заключение

В работе рассмотрена применимость метода AIPS, в рамках двухступенчатого предобуславливателя CPR. Предложен ориентированный на архитектуру CUDA параллельный алгоритм решения линейных систем с трехдиагональной матрицей специального вида: состоящей из независимых блоков различного размера.

- Разработанный алгоритм для решения СЛАУ с трехдиагональной матрицей позволяет получить ускорение более чем в 2 раза относительно методов из библиотеки cuSPARSE.
- CPR-AIPS позволяет снизить время решения большинства тестовых матриц относительно CPR-AMG на 1 GPU, а также показывает приемлемую масштабируемость до 2 раз на 2 GPU V100 и до 3,6 на 4 GPU.
- Полученные результаты создают предпосылки для успешного применения CPR-AIPS в качестве высокопроизводительного предобуславливателя на современных гибридных вычислительных системах, оснащенных GPU NVIDIA.

Вопросы?