

USPARS – параллельный прямой решатель разреженных СЛАУ

Новосибирский центр информационных технологий
УНИПРО

23 ноября 2023

Д. Быков, В. Костин, С. Соловьев



Прямые решатели разреженных СЛАУ

- ❑ Основа математического моделирования
- ❑ Компонент CAE (Computer-Aided Engineering)

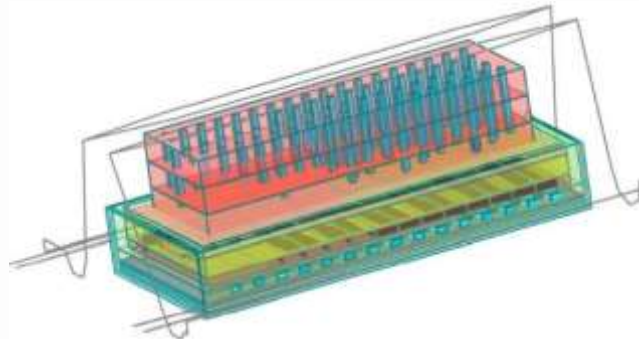
Программные пакеты:

- ❑ UMFPACK (USA)
- ❑ PARDISO (Switzerland)
- ❑ **MKL PARDISO (USA)**
- ❑ **MUMPS (France)**
- ❑ SuperLU (USA)
- ❑ USPARS (Россия)

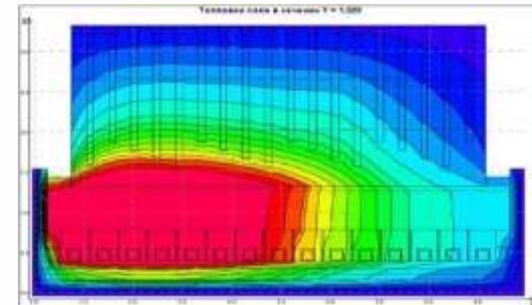
Нефтяная промышленность:
Морская сейсморазведка



Оптимизация токораспределения



Прогнозирование
аварийных случаев



Краткое описание архитектуры

$$Ax = b$$

$$A' = D_l P P_m A P^t D_r$$

P – матрица перестановок, минимизирующая заполняемость ненулевыми элементами треугольных факторов (fill-in)

P_m – перестановка больших элементов A на главную диагональ (matching);

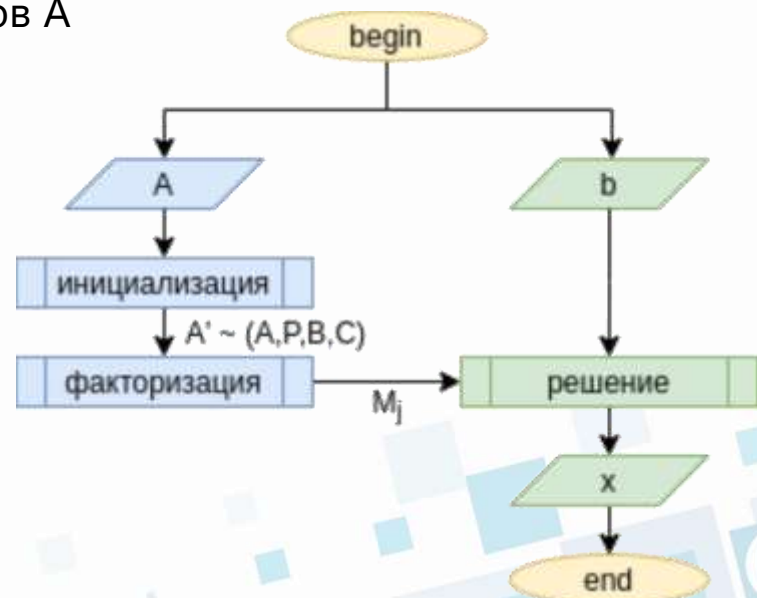
D_l, D_r - диагональные матрицы: масштабирование элементов A (scaling).

$$A' = \{LL^H | LDL^H | LDL^T | LU\},$$

L – нижнетреугольная;

U – верхнетреугольная;

D – диагональная.



Уменьшения заполненности факторов (fill-in)

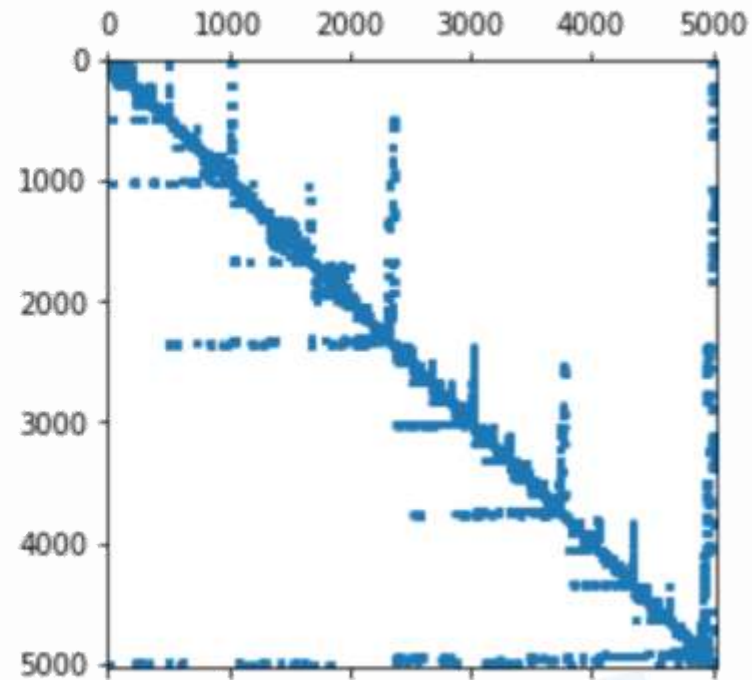
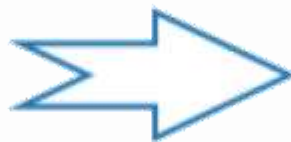
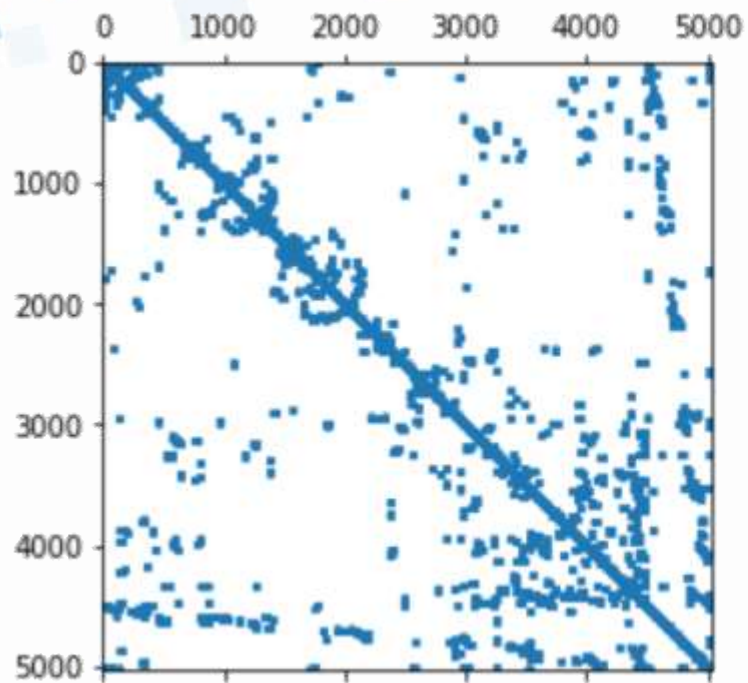
$$A = \begin{pmatrix} 9 & 1.5 & 6 & 0.75 & 3 \\ 1.5 & 0.5 & & & \\ 6 & & 12 & & \\ 0.75 & & & 0.625 & \\ 3 & & & & 16 \end{pmatrix} = LL^t$$

$$L = \begin{pmatrix} 3 & & & & \\ 0.5 & 0.5 & & & \\ 2 & -2 & 2 & & \\ 0.25 & -0.25 & -0.5 & 0.5 & \\ 1 & -1 & -2 & -3 & 1 \end{pmatrix}$$

$$A' = PAP^t = \begin{pmatrix} 16 & & & & 3 \\ & 0.5 & & & 0.75 \\ & & 12 & & 6 \\ & & & 0.625 & 1.5 \\ 3 & 1.5 & 6 & 0.75 & 9 \end{pmatrix} = LL^t$$

$$L = \begin{pmatrix} 4 & & & & \\ & \frac{1}{\sqrt{2}} & & & \\ & & 2\sqrt{3} & & \\ & & & \frac{\sqrt{10}}{4} & \\ \frac{3}{4} & \frac{3}{\sqrt{2}} & \sqrt{3} & \frac{3}{\sqrt{10}} & \sqrt{\frac{3}{80}} \end{pmatrix}$$

Примеры портретов $A \rightarrow A'$



USPARS – прямой решатель СЛАУ

Реестровая запись №18945 от 05.09.2023

<https://unipro.ru/uspars/>

Предпосылки создания пакета:

- Не было готового продукта для отечественных CPU (**Эльбрус, Байкал**)
- Наличие команды, обладающей **экспертизой в разработке прямых решателей** и сопутствующих алгоритмов, позволяло подумать о конкурентоспособности не только в России, но и в мире
- Возможность предоставления быстрой качественной **поддержки для российских пользователей**, помощь в подборе параметров решателя
- Возможность **полного контроля над исходным кодом пакета**.
- Выгоднее иметь свой собственный продукт, который стоит дешевле имеющихся на рынке и который можно настраивать и расширять под свои нужды

Состав USPARS

Функциональное наполнение - классическое для прямых решателей разреженных матриц:

- ❑ Полный набор функций для осуществления треугольной факторизации и решения СЛАУ: **LU, LDLT, LDLH, LLH** разреженных матриц с вещественными и комплексными элементами одинарной и двойной точности
- ❑ **Собственная компонента** (Atlant) **переупорядочивания элементов** матрицы для уменьшения заполненности факторов и улучшения параллелизации
- ❑ Средства повышения численной устойчивости процесса решения (scaling, boosting и global pivoting)
- ❑ Возможность использования итерационного уточнения для улучшения точности решения

Оптимизация производительности для серверов с общей памятью на основе многоядерных процессоров:

- ❑ **Все стадии** решения USPARS **распараллелены** средствами библиотеки OpenMP и базируются на использовании общеупотребимых функций высокооптимизированных библиотек (BLAS).

PyUspars: SciPy-совместимый Python-интерфейс:

- ❑ Простой (близкий к естественному) синтаксис задания запросов на решение
- ❑ Автоматический контроль за выделением, распределением и освобождением динамической памяти
- ❑ Использование **стандартного** для Python **Sparse CSR** формата матриц

Python-пакет. Преимущества

Удобный пользовательский интерфейс:

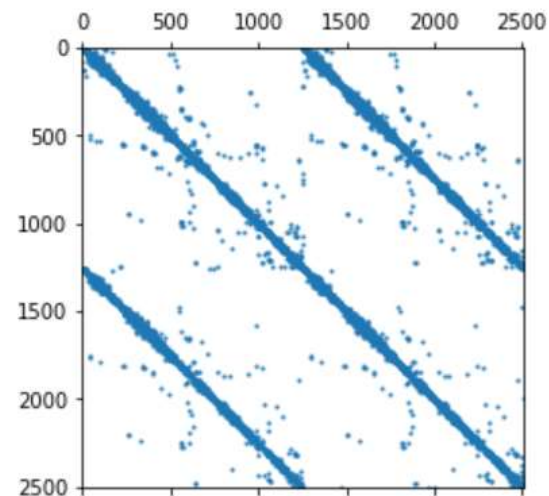
- ❑ **Простой** (близкий к естественному) **синтаксис** задания запросов на решение
- ❑ **Автоматический контроль** за выделением, распределением и освобождением динамической памяти

Расширенный функционал, направленный на верификацию полученного решения:

- ❑ Расчет невязки, погрешности
- ❑ Визуализации исходной матрицы, решения, перестановок
- ❑ Нахождение числа обусловленности матрицы
- ❑ Используется в качестве инструмента для написания и тестирования нового функционала

Пример решения системы с визуализацией матрицы:

```
A = mmread("1.mtx").tocsr()
plt.spy(A, markersize=1)
x_exact = [1]*A.shape[0]
b = A*x_exact
```



```
tt = PyUspars()
tt.init(A, ftype.LU)
tt.fact()
x = tt.solve(b)
print("Error -", tt.calc_norm_error(x_exact))
print("Discrepancy -", tt.calc_norm_residual())
```

```
Error - 1.4934960850244645e-12
Discrepancy - 7.13933003711876e-14
```


Python-пакет. SciPy-совместимость

- ❑ Полная совместимость со стандартным для **Python Sparse CSR** форматом матриц
- ❑ Доступны все встроенные в SciPy оптимизированные операции над разреженными матрицами
- ❑ Возможность использования совместимых с этим форматом визуализаций, методов работы с данными, чтения и записи
- ❑ **Совместимость интерфейсов** прямых разреженных решателей, позволяющая использовать Uspars вместо стандартных методов SciPy путем замены импорта.

```
from pyuspars_package.solver import PyUspars
from pyuspars_package.solver import FACTORIZE_TYPE as ftype
from pyuspars_package.scipy import factorized, spsolve
import numpy as np
from scipy import sparse
from scipy.io import mmread
import matplotlib.pyplot as plt
```

Замена импорта:

```
from scipy.sparse import spsolve
from pyuspars_package.scipy import spsolve
```

Статус работ

- ❑ Uspars оттестирован в плане точности, надежности и производительности на **наборе тестовых матриц**, взятых с ресурсов <https://sparse.tamu.edu/> (**800 матриц**), <http://www.elses.jp/matrix/>, а так же матрицах пользователей
- ❑ Надежность, точность и производительность Uspars не уступают, а во многих случаях и **превосходят показатели** для зарубежных конкурентов.
- ❑ Uspars работает на платформах **x86, Эльбрус, ARM**.
- ❑ Оптимизирована производительность для серверов с общей памятью на основе **многоядерных процессоров**:
 - Параллелизация средствами OpenMP.
 - Использование высоко-оптимизированных математических библиотек.
 - Ряд авторских алгоритмических оптимизаций.

Пользователи USPARS:

- ❑ Коммерческие – **ФГУП РФЯЦ-ВНИИЭФ** (ЛОГОС Прочность, Логос ЭМИ)
- ❑ Некоммерческие – **ССКЦ** (Сибирский суперкомпьютерный центр СО РАН)

Результаты сравнительных прогонов: USPARS/PARDISO/MUMPS

- ❑ Решение СЛАУ с одной правой частью сгенерированной по заданному вектору решений
- ❑ Матрицы:
 - Suite Sparse Matrix Collection (<https://sparse.tamu.edu/>)
 - 24 симметричных и 12 несимметричных
 - Размер СЛАУ 14 000 ... 5 500 000 неизвестных
- ❑ Система:

№1 (AMD, 128 ядер)	№2 (Intel, 24 ядра)	№3 (Эльбрус, 32 ядра)
<ul style="list-style-type: none">▪ 2 процессора AMD EPYC 7763 64-Core▪ Оперативная память 1024 Gb▪ Aramid 3.1 (Red Hat Enterprise Linux 8.3)	<ul style="list-style-type: none">▪ 2 процессора Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz (12 core, HT disabled)▪ 256 GB RAM▪ Ubuntu 22.04	<ul style="list-style-type: none">▪ 2 процессора E16C (16 ядер) (http://mcst.ru/Elbrus-16C)▪ 512 GB RAM▪ OS Elbrus 7.1

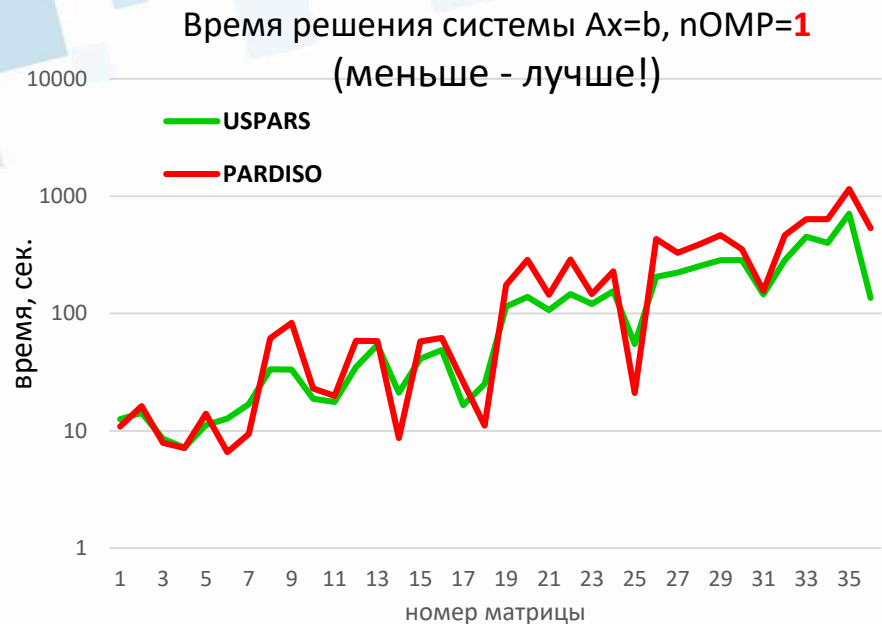
- ❑ Проверяемые характеристики:
 - Точность решения
 - Размер используемой памяти
 - Производительность

USPARS/PARDISO/MUMPS: Список матриц

#	имя матрицы	размер	число ненулевых	число обусловленности
1	apache2	715 176	2 766 523	3,05E+06
2	torso3	259 156	4 429 042	2,27E+02
3	c-big	345 241	1 343 126	4,05E+04
4	ldoor	952 203	23 737 339	1,69E+08
5	F1	343 791	13 590 452	3,52E+06
6	thermal2	1 228 045	4 904 179	4,30E+06
7	G3_circuit	1 585 478	4 623 152	1,39E+07
8	H2O	67 024	1 141 880	4,24E+03
9	sparsine	50 000	799 494	2,78E+06
10	af_shell10	1 508 065	27 090 195	1,41E+10
11	boneS10	914 898	28 191 660	1,03E+08
12	appu	14 000	1 853 104	1,71E+02
13	PFlow_742	742 793	18 940 627	1,02E+12
14	memchip	2 707 524	14 810 202	1,41E+07
15	dielFilterV3real	1 102 824	45 204 422	8,67E+06
16	dielFilterV2real	1 157 456	24 848 204	8,91E+06
17	TSOPF_RS_b2383_c1	38 120	16 171 169	2,20E+08
18	Freescale1	3 428 755	18 920 347	1,03E+10

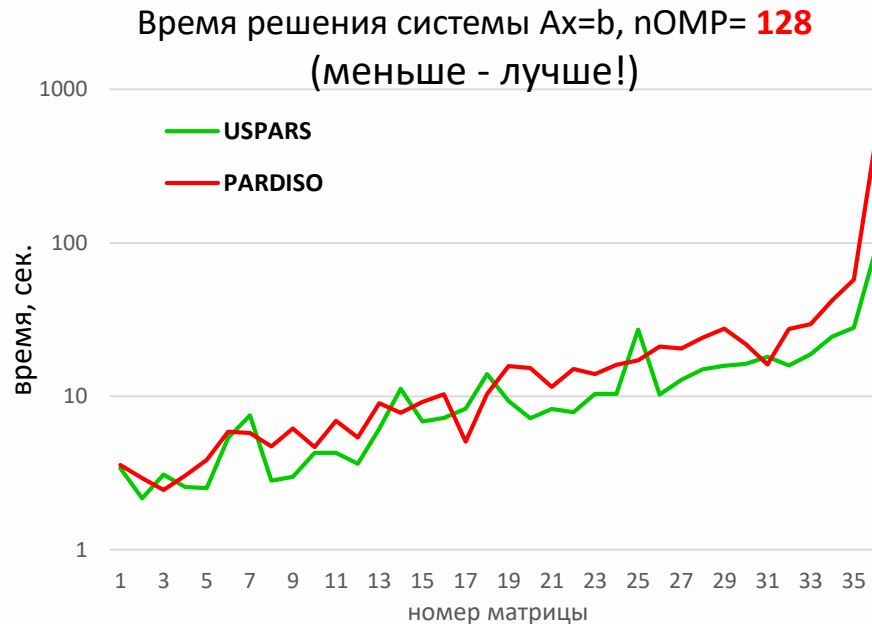
#	имя матрицы	размер	число ненулевых	число обусловленности
19	StocF-1465	1 465 137	11 235 263	2,41E+13
20	Ga10As10H30	113 081	3 114 357	2,90E+05
21	bone010	986 703	36 326 514	4,42E+08
22	Ge99H100	112 985	4 282 190	6,40E+03
23	Flan_1565	1 564 794	59 485 419	1,21E+08
24	audikw_1	943 695	39 297 771	1,83E+10
25	rajat31	4 690 002	20 316 253	3,33E+06
26	Ga19As19H42	133 123	4 508 981	3,07E+06
27	Hook_1498	1 498 023	31 207 734	3,60E+06
28	atmosmodl	1 489 752	10 319 760	1,05E+03
29	atmosmodd	1 270 432	8 814 880	4,88E+03
30	Transport	1 602 111	23 500 731	1,23E+06
31	ML_Geer	1 504 002	110 879 972	6,32E+08
32	atmosmodj	1 270 432	8 814 880	6,04E+03
33	Geo_1438	1 437 960	32 297 325	1,14E+13
34	vas_stokes_1M	1 090 664	34 767 207	1,73E+07
35	Serena	1 391 349	32 961 525	1,11E+14
36	circuit5M	5 558 326	59 524 291	3,43E+10

USPARS vs. PARDISO: производительность (AMD, 128 ядер)



Среднее геометрическое отношений времен
(Geomean)
больше единицы-лучше!

1,24

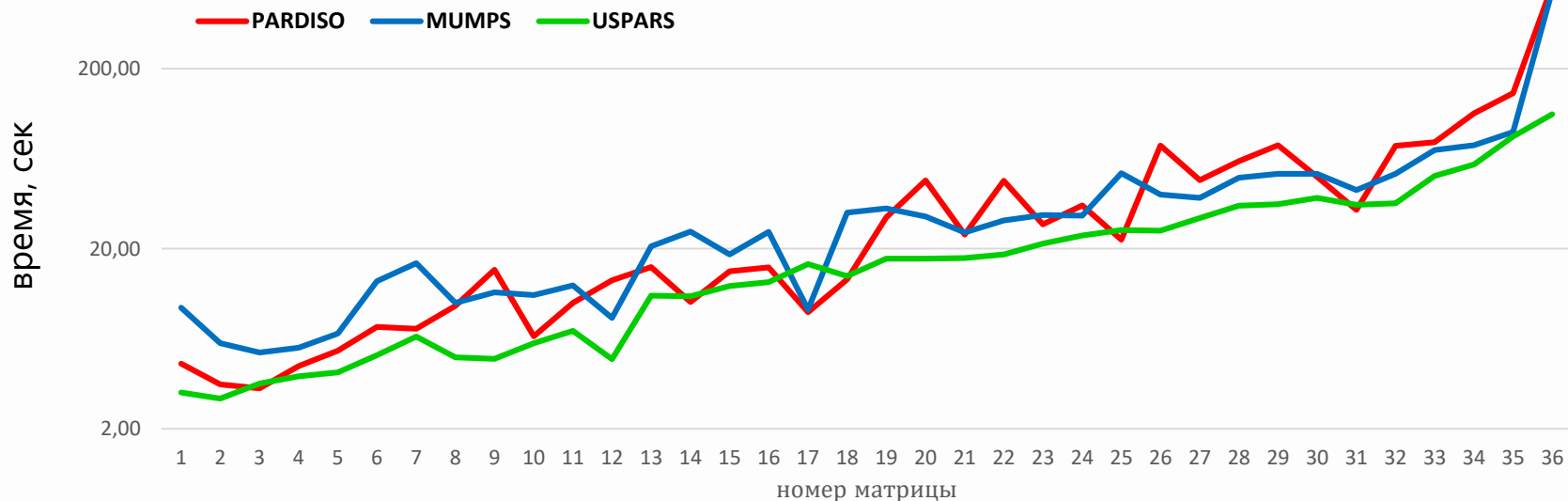


Среднее геометрическое отношений времен
(Geomean)
больше единицы-лучше!

1,38

USPARS/PARDISO/MUMPS: производительность (Intel, 24 ядра)

Время решения системы $Ax=b$, nOMP=24
(меньше - лучше!)

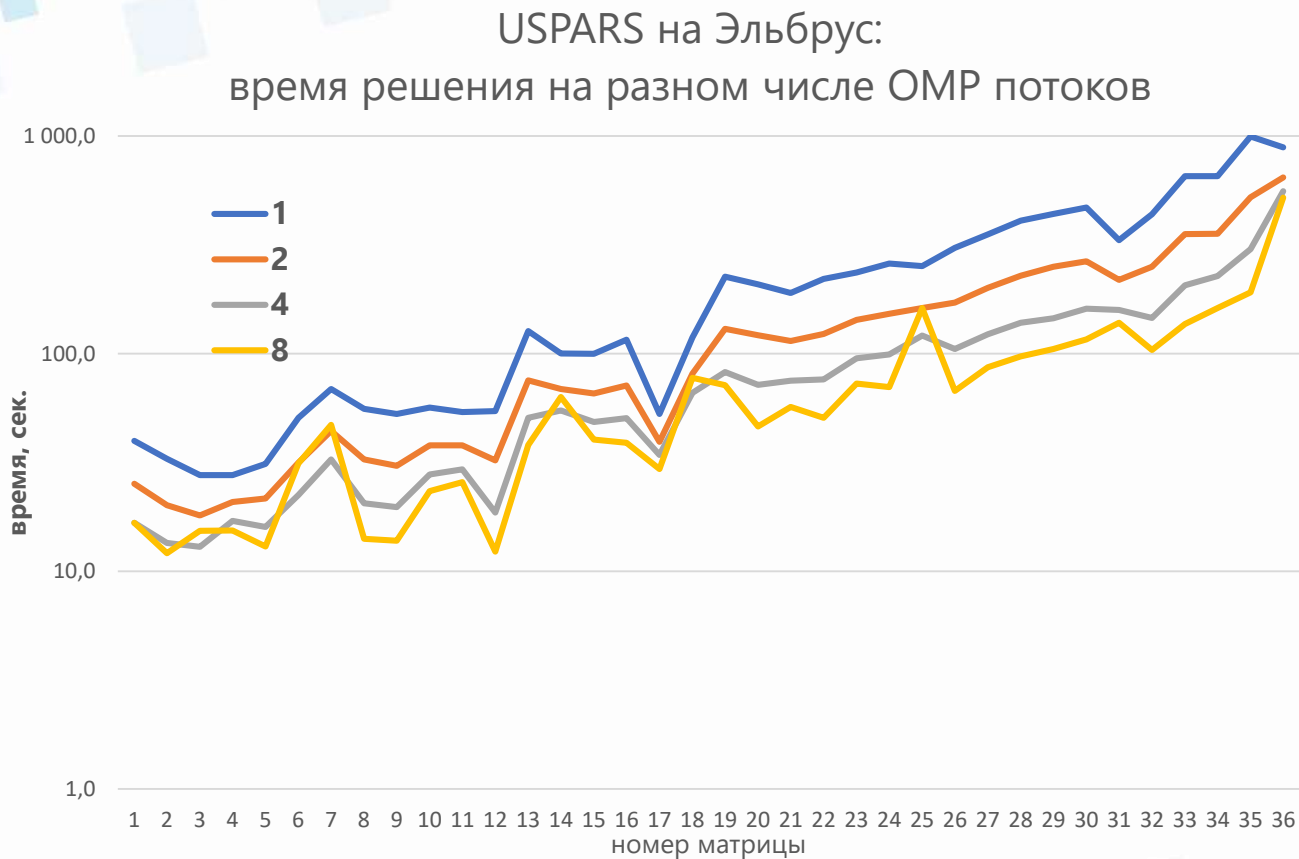


Среднее геометрическое отношений времен (Geomean)
больше единицы-лучше!

1,52 (vs PARDISO)

1,68 (vs MUMPS)

USPARS: Производительность на Elbrus



Elbrus

nOMP	Time ratio: 1OMP/nOMP	
	Min	Max
1	1,00	1,00
2	1,33	1,90
4	1,54	3,29
8	1,46	5,20
24	0,87	6,32

x86

nOMP	Time ratio: 1OMP/nOMP	
	Min	Max
1	1,00	1,00
2	1,11	1,77
4	1,48	3,57
8	1,62	6,80
24	1,62	13,93

Подключение модуля на ССКЦ ИВМиМГ СО РАН

- ❑ Подключение модуля, содержащего актуальный USPARS
`module load uspars/latest`
- ❑ Подключение модуля, содержащего USPARS определенной версии
 - `module load uspars/2.2.2/mkl` – USPARS 2.2.2 с математической библиотекой Intel MKL
 - `module load uspars/2.2.2/ublas` – USPARS 2.2.2 с собственной оптимизированной математической библиотекой UBLAS
- ❑ Примеры, руководство пользователя, Python-пакет доступны в общедоступной директории
`/opt/software/applied/unipro/uspars/`

Сборка примеров на ССКЦ ИВМиМГ СО РАН

1. Подключение модуля

```
module load uspars/latest
```

2. Копирование примеров и заголовочного файла USPARS в домашнюю директорию

```
cp -r ${USPARS_PATH}/examples ~/uspars_examples
```

```
cp -r ${USPARS_PATH}/headers/uspars.h ~/uspars_examples
```

3. Сборка примеров

```
cd ~/uspars_examples; bash make_examples.sh
```

4. Запуск примеров

```
bash run_examples.sh
```

5. Вывод запусков будет записан в директорию log

Структура запуска USPARS

на примере example_LU_d.c

1. Задание скалярного типа и типа факторизации

```
usp_scalar_type utype = USP_DOUBLE;  
usp_factorize_type ftype = USP_LU;
```

2. Задание опций решателя – отключение логирования

```
int options[USP_OPTIONS];  
for (i = 0; i < USP_OPTIONS; i++)  
    options[i] = -1;  
options[USP_MSGLVL] = 0;
```

3. Инициализация

```
ier = uspars_init(&tt, N, ia, ja, a, utype, ftype, options);
```

4. Факторизация

```
ier = uspars_fact(&tt);
```

5. Решение

```
ier = uspars_solve(&tt, nrhs, x, utype);
```

6. Вектор x – итоговый ответ в задаче

Дорожная карта (2023)

Расширение функциональности:

- ❑ Частичная **проблема собственных значений** для разреженной вещественной симметричной или комплексной эрмитовой матрицы.
 - Вычисляются собственные значения, принадлежащие заданному интервалу и соответствующие собственные векторы.
 - Будет использован известный алгоритм FEAST, в котором в качестве решателя систем с комплексными симметричными матрицами используется USPARS.
- ❑ Версия решателя для серверов с возможностью **хранения промежуточных данных на диске** (ООС версия)
 - Позволяет увеличивать размеры решаемых СЛАУ
 - Замедляет процесс вычислений из-за обменов данными с диском
- ❑ Поддержка **long long int** интерфейсов для решения систем с числом ненулевых элементов, больших 2.147.483.647

Дорожная карта (2024-...)

Полноценная поддержка архитектур Эльбрус и ARM

Расширение функциональности:

- ❑ Решение СЛАУ с **прямоугольными** разреженными матрицами коэффициентов
- ❑ Методы регуляризации для решения СЛАУ с **плохо обусловленными** разреженными матрицами коэффициентов
- ❑ Использование **малоранговой аппроксимации** для сжатия промежуточных данных, возникающих в процессе факторизации матрицы
 - Снижает нагрузку на память, уменьшает число требующихся арифметических операций;
 - Для исключения влияния аппроксимации на точность решения требует включения в процесс вычислений итерационного уточнения.

Разработка версии решателя для вычислительных систем с **распределенной памятью** (кластеров):

- ❑ Символьная факторизация
- ❑ Численная факторизация
- ❑ Этап решения

Спасибо за внимание!



<https://unipro.ru/uspars/>

Тел: +7 (383) 373 2463

Почта: marketing@unipro.ru

**Новосибирский Центр Информационных Технологий
«УНИПРО»**