



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МАТЕМАТИКИ И МЕХАНИКИ

КАФЕДРА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ И СИСТЕМНОГО ПРОГРАММИРОВАНИЯ

ЛАБОРАТОРИЯ ITLav



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МАТЕМАТИКИ И МЕХАНИКИ
КАФЕДРА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ И СИСТЕМНОГО ПРОГРАММИРОВАНИЯ
ЛАБОРАТОРИЯ ITLAB

***ВВЕДЕНИЕ В АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ
И ОПТИМИЗАЦИЮ ПРОГРАММ***

**Прямое решение СЛАУ с разреженной матрицей
на мини-кластере с процессорами архитектуры RISC-V**



**LOBACHEVSKY
UNIVERSITY**

Пирова А.Ю., Мееров И.Б.

Содержание

- ❑ Цель лекции
- ❑ Предметная область
- ❑ Прямое решение СЛАУ с разреженной матрицей
- ❑ ПО для переупорядочения и решения СЛАУ
- ❑ Методика тестирования
- ❑ Результаты вычислительных экспериментов на процессорах x86, RISC-V

Цели лекции

- ❑ Продемонстрировать возможности решения разреженных СЛАУ готовыми программными пакетами на процессорах архитектуры RISC-V.
- ❑ Показать анализ и сравнение производительности на процессорах архитектур RISC-V и x86.

Предметная область

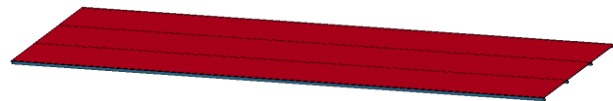
- При выполнении численного моделирования конечно-элементными методами часто возникает задача решения системы линейных алгебраических уравнений (СЛАУ) с разреженной матрицей большого порядка.



модель лопатки
авиационного двигателя
Матрица 2.5 млн строк,
174 млн. элементов



модель корпуса
опоры авиадвигателя
Матрица 2.5 млн строк,
184 млн. элементов



модель кессонов самолета
Матрица 4.2 млн строк,
285 млн. элементов

Прямое решение СЛАУ с разреженной матрицей...

- Пусть дана система линейных алгебраических уравнений:

$$Ax = b$$

A – симметричная разреженная матрица, b – плотный вектор,
 x – вектор неизвестных.

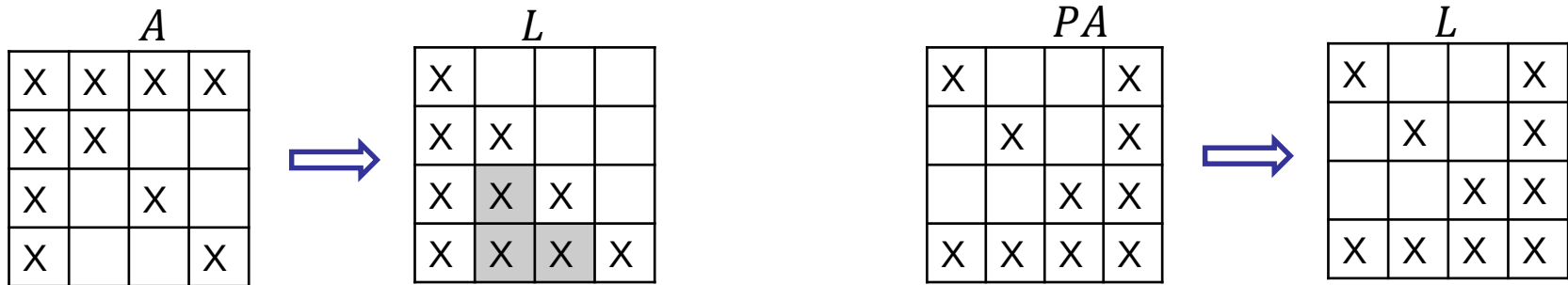
- Методы решения СЛАУ – *прямые и итерационные*
- Прямой метод решения – разложение, в зависимости от типа матрицы:
 - $A = LL^T$ (разложение Холецкого) – матрица симметричная положительно определенная;
 - $A = LDL^T$ – матрица симметричная неопределенная;
 - $A = LU$ – матрица общего вида.

- Далее рассмотрим разложение вида $A = LDL^T$

Прямое решение СЛАУ с разреженной матрицей...

- При прямом решении СЛАУ с разреженной матрицей возникает проблема **заполнения** – существенного увеличения числа ненулевых элементов фактора матрицы.
- Уменьшить заполнение можно с помощью симметричной перестановки:

$$Ax = b \iff (PA P^T)(Px) = Pb, P - \text{матрица перестановки}$$



- Задача нахождения оптимальной перестановки **NP-трудная** (Yannakakis, 1981), решается эвристическими методами (метод вложенных сечений, метод минимальной степени и др.)

Прямое решение СЛАУ с разреженной матрицей

□ Процедура прямого решения разреженной СЛАУ с симметричной матрицей:

Фаза анализа



Численная фаза



Обратный ход



Постобработка решения

- Масштабирование значений элементов матрицы A ;
- Нахождение перестановки P , минимизирующей размер фактора, $A'' = PAP$;
- Символьная факторизация.

- Численное разложение $A'' = LDL^T$

- Решение треугольных систем $LDy = Pb, L^T x = y$

- Обратная перестановка вектора x ;
- Итерационное уточнение решения x ;
- Анализ вычислительных ошибок.

ПО для переупорядочения и решения СЛАУ...

- Библиотеки для переупорядочения разреженных матриц:
 - **ParMETIS** – переупорядочиватель разреженных матриц с открытым исходным кодом (университет Миннесоты)
 - многоуровневый метод вложенных сечений;
 - параллелизм для распределенной памяти, технология MPI.
 - **PT-Scotch** - переупорядочиватель разреженных матриц с открытым исходным кодом (LaBRI, университет Бордо)
 - многоуровневый метод вложенных сечений;
 - параллелизм для распределенной памяти, технология MPI.
 - **DMORSy** – переупорядочиватель разреженных матриц, собственная разработка ННГУ;
 - многоуровневый метод вложенных сечений;
 - гибридная MPI + OpenMP схема параллелизма.

ПО для переупорядочения и решения СЛАУ

- ❑ Библиотеки для решения СЛАУ с разреженной матрицей:
 - MUMPS, SuperLU, CHOLMOD – академические разработки, свободно распространяемые решатели СЛАУ с открытым кодом;
 - Intel oneAPI MKL PARDISO - решатель, оптимизированный для процессоров Intel, свободно распространяемый, с закрытым кодом.

- ❑ **MUMPS** широко используется в академической среде.
 - гибридная MPI + OpenMP схема параллелизма, используется BLAS.
 - Есть интерфейс для интеграции с METIS, ParMETIS, Scotch, PT-Scotch;
 - Есть возможность загрузить пользовательскую перестановку.

Методика тестирования

□ Протестируем:

- переупорядочиватели ParMETIS и DMORSy. Сравним их время работы и размер фактора матриц после перестановки.
- Решатель MUMPS с разными переупорядочивателями. Сравним общее время решения СЛАУ, размер фактора матриц после перестановки.

□ Цели тестирования:

- Проверить работоспособность переупорядочивателей и решателя MUMPS на процессорах архитектуры RISC-V без изменений исходного кода.
- Сравнить соотношение производительности ParMETIS, DMORSy, MUMPS на процессорах архитектур x86 и RISC-V. Сохраняются ли тенденции, наблюдающиеся на x86, на процессорах RISC-V?

Тестовое окружение

□ RISC-V:

- мини-кластер из четырех 4-ядерных процессоров CPU TH1520, частотой 1.848Ghz, память 8 GB, операционная система Debian 12.
- компиляторы gcc v. 13.2.0, gfortran; решатель MUMPS v. 5.6.2, библиотеки OpenBLAS v. 0.3.26 и ParMETIS v. 4.0.3.

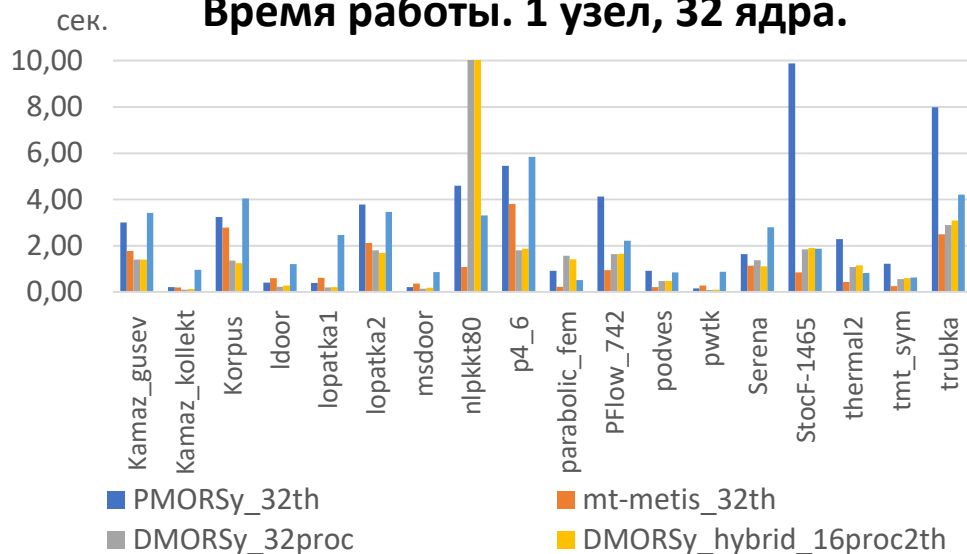
□ x86:

- узлы кластера МЦЦ РАН: процессор Intel Xeon Gold 6154 (Skylake), 3.0 GHz, 2 x 18 ядер, память 192 GB, компилятор Intel Parallel Studio XE 2017 Cluster Edition.
- решатель MUMPS v. 5.4.1, MKL BLAS

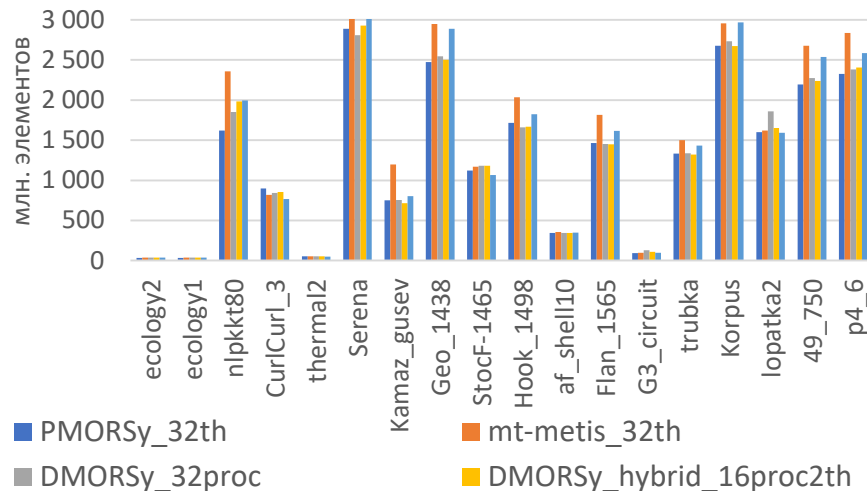
□ **Тестовые матрицы:** симметричные матрицы из коллекции SuiteSparse (<https://sparse.tamu.edu/>) порядком от 0.5 млн до 1.5 млн строк и заполнением порядка $1 \times 10^{-6} - 1 \times 10^{-5}$ ненулевых элементов.

Сравнение переупорядочивателей, x86

Время работы. 1 узел, 32 ядра.



Заполнение фактора. 1 узел, 32 ядра

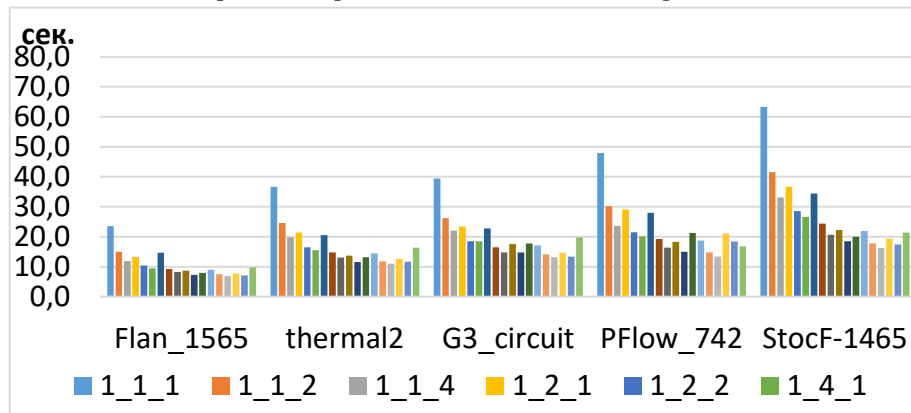


- DMORSy работает быстрее, чем ParMETIS, на матрицах порядка менее 1 млн. строк и матрицах с регулярной структурой.
- По заполнению фактора на половине тестовых матриц DMORSy опережает ParMETIS на ~8%, на других матрицах DMORSy уступает ParMETIS на ~12%.

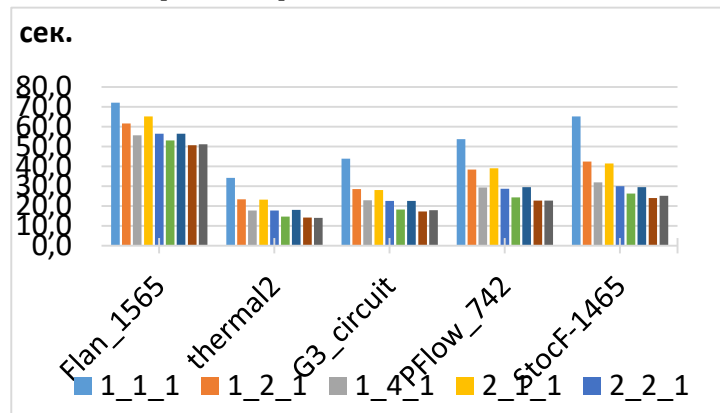
Сравнение переупорядочивателей, RISC-V

- Параметры переупорядочивателей фиксированы.
- Подбор наилучшего сочетания числа процессов и потоков:
 - Для DMORSy – 4 узла x (1 процесс x 4 потока), среднее ускорение – 2.8 раз;
 - Для ParMETIS – 4 узла x (2 или 4 процесса x 1 поток), среднее ускорение – 1.8 раз

Время работы DMORSy



Время работы ParMETIS

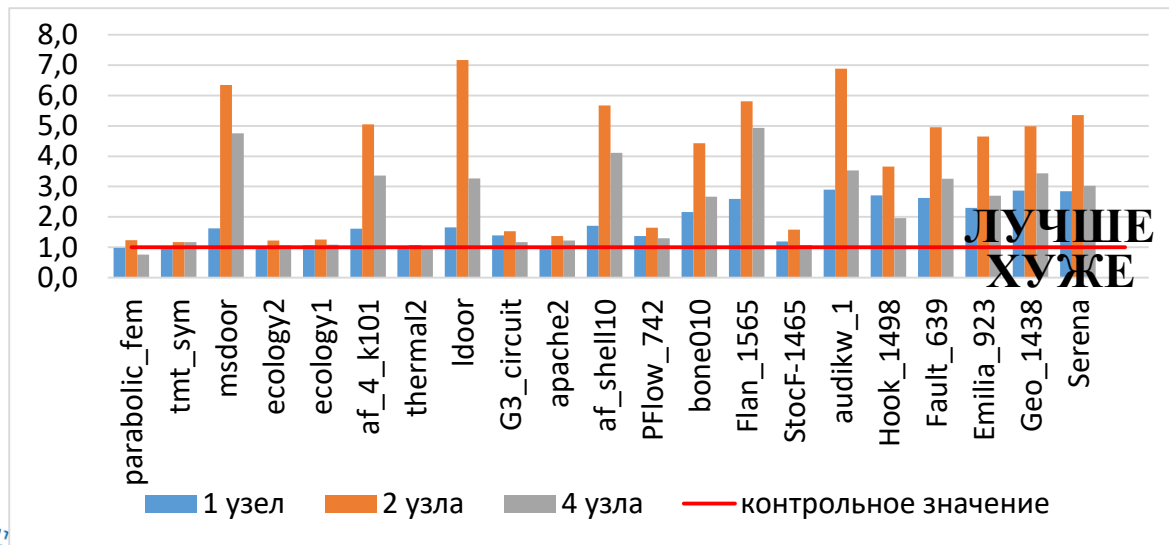


X_Y_Z, где X – число узлов, Y – число процессов на узел, Z – число потоков на процесс

Сравнение переупорядочивателей, RISC-V

- ❑ Сравнение DMORSy и ParMETIS по двум критериям: заполнение фактора после применения перестановки (чем меньше, тем лучше), время работы.
- ❑ Параметры DMORSy, дающие наименьшее заполнение.

Время работы ParMETIS относительно DMORSy, разы

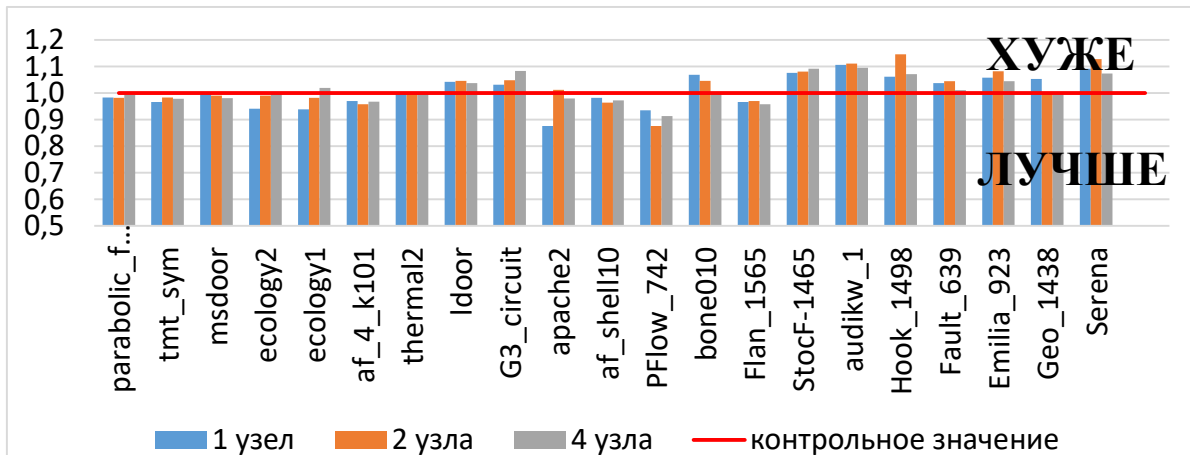


- В среднем, на двух узлах DMORSy работает быстрее ParMETIS в 3,7 раз, на четырех узлах – в 2,4 раза.
- ParMETIS плохо масштабируется, на 2/3 тестовых матриц ускоряется в среднем в 1,8 раз на 4х узлах.
- Среднее ускорение DMORSy 1,6 раз на 4х узлах.

Сравнение переупорядочивателей, RISC-V

- ❑ Сравнение DMORSy и ParMETIS по двум критериям: заполнение фактора после применения перестановки (чем меньше, тем лучше), время работы.
- ❑ Параметры DMORSy, дающие наименьшее заполнение.

Заполнение фактора, полученное DMORSy, относительно ParMETIS, разы

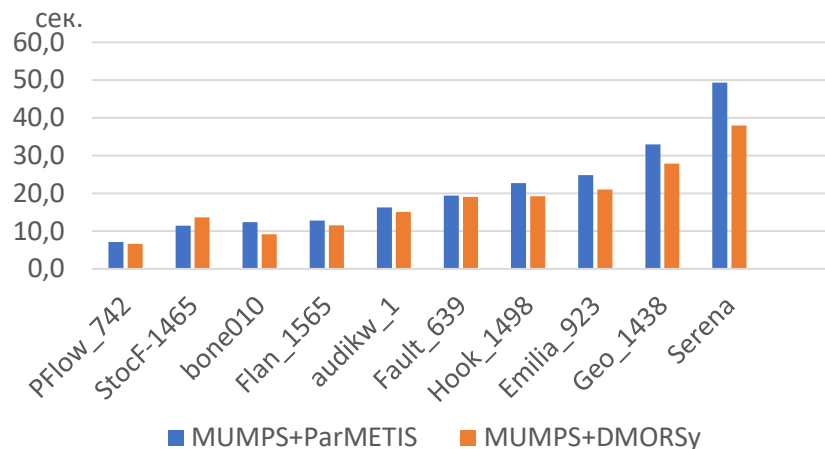


Близкое заполнение фактора DMORSy и ParMETIS:

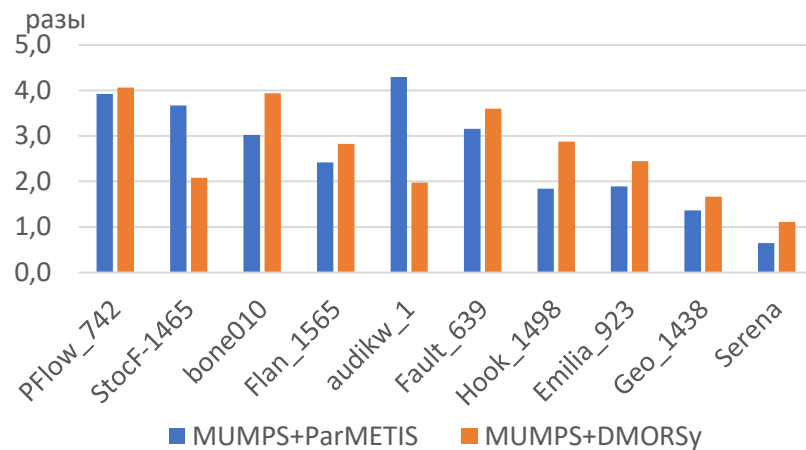
- на половине тестовых матриц DMORSy опережает ParMETIS на ~3%.
- на других матрицах DMORSy уступает ParMETIS на ~6%.
- DMORSy дает лучшие перестановки на более разреженных матрицах.

Вычислительные эксперименты для решателя СЛАУ, x86

Время работы MUMPS на 1 узле кластера МСЦ, 32 ядра



Ускорение MUMPS относительно последовательной версии, 32 ядра

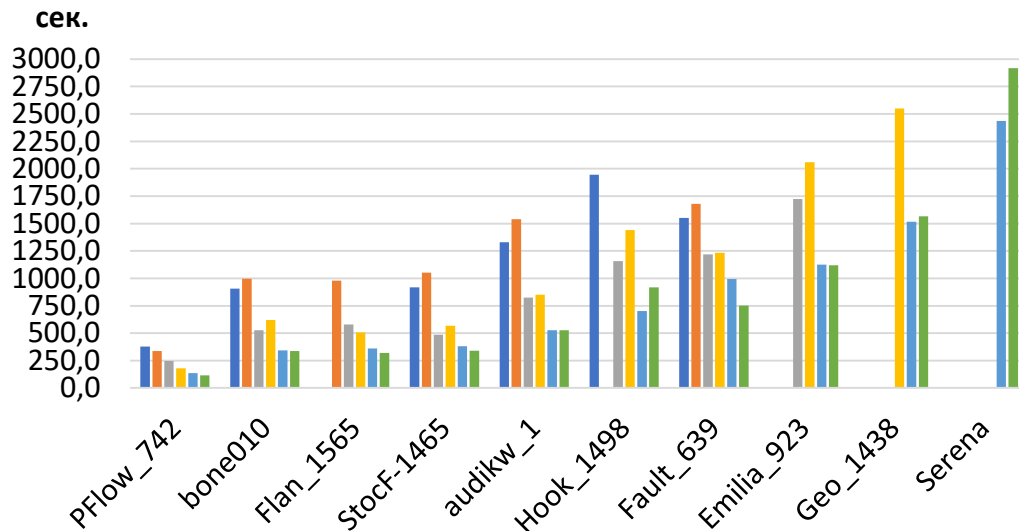
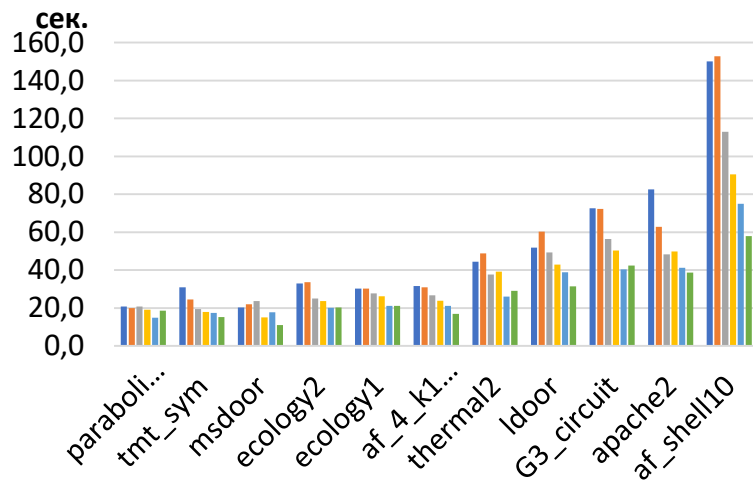


На большинстве тестовых матриц MUMPS с DMORSy работал быстрее, чем с ParMETIS, в среднем – на 26%. Опережение получено за счет сокращения времени переупорядочения и времени численной факторизации.

Среднее ускорение MUMPS – 2,8 раз.

Вычислительные эксперименты для решателя СЛАУ, RISC-V

Время работы решателя MUMPS, 1 – 4 узла x (4 ядра)



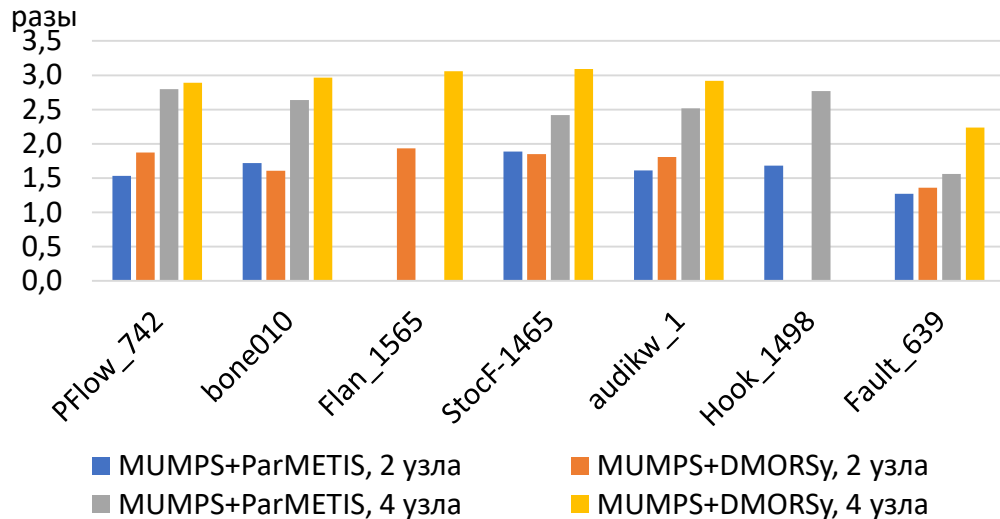
■ MUMPS+ParMETIS 1 узел
■ MUMPS+DMORSy 2 узла

■ MUMPS+DMORSy 1 узел
■ MUMPS+ParMETIS 4 узла

■ MUMPS+ParMETIS 2 узла
■ MUMPS+DMORSy 4 узла

Вычислительные эксперименты для решателя СЛАУ, RISC-V

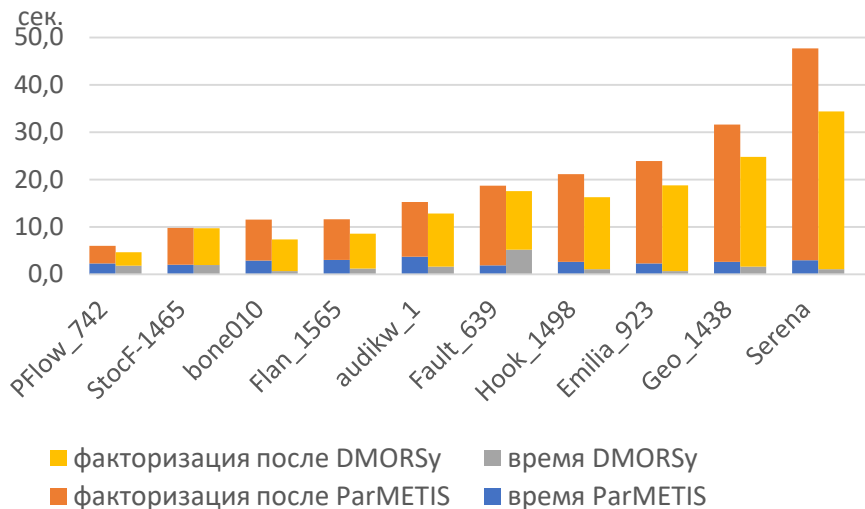
Ускорение решателя MUMPS относительно работы на одном узле



Масштабируемость общего времени решения СЛАУ с DMORSy лучше, чем с ParMETIS. Среднее ускорение на 4х узлах для больших матриц: MUMPS+DMORSy – 2,8 раз, MUMPS+ParMETIS – 2,4 раз.

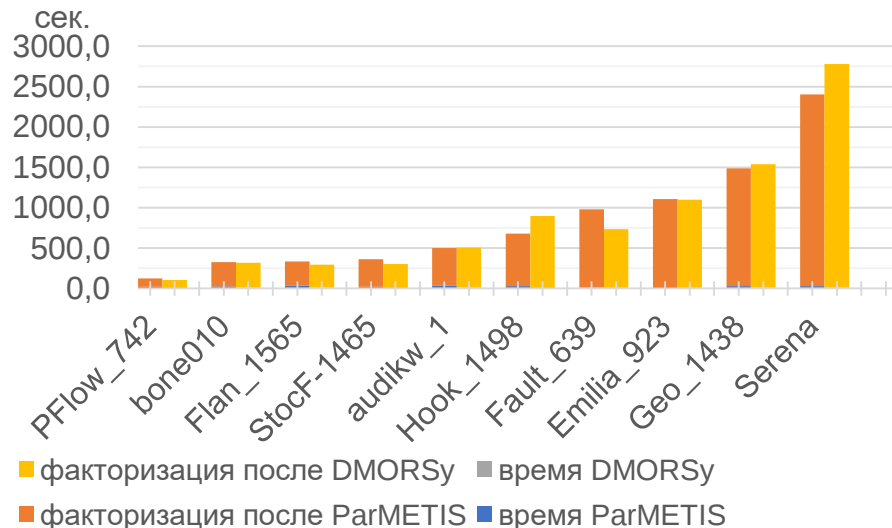
Вычислительные эксперименты для решателя СЛАУ, RISC-V

Соотношение времени
переупорядочения и времени решения
СЛАУ, 32 ядра, x86



Время факторизации в ~4–30 больше,
чем время переупорядочения

Соотношение времени
переупорядочения и времени решения
СЛАУ, RISC-V, 16 ядер



Время факторизации до 100 раз больше,
чем время переупорядочения

Выводы из результатов экспериментов

- ❑ Все рассматриваемые библиотеки работают на RISC-V без изменения исходных кодов.
- ❑ Объем памяти одного вычислительного узла достаточно для получения перестановки на матрицах порядка до $2,5 \times 10^6$ строк и $\sim 2,3 \times 10^7$ ненулевых элементов, решение СЛАУ – на матрицах с размером фактора до $\sim 1,7 \times 10^9$ ненулевых элементов.
- ❑ На кластере из устройств Lichee Pi 4A целесообразно использование гибридной MPI+OpenMP схемы распараллеливания.
- ❑ В сравнении с результатами решения СЛАУ на процессорах архитектуры x86, на RISC-V наблюдается значительное замедление численной фазы решения.
- ❑ Потенциал для сокращения времени работы решателя на RISC-V – использование оптимизированной библиотеки базовых алгоритмов линейной алгебры (BLAS).

Литература...

1. Davis T.A. Direct methods for sparse linear systems. – Society for Industrial and Applied Mathematics, 2006. – 230 p.
2. MULTifrontal Massively Parallel Solver (MUMPS 5.6.1) User's guide // Tech. rep. ENSEEINT-IRIT. – 2023. URL: <https://mumps-solver.org/index.php?page=doc>
3. Karipis, G. ParMETIS. Parallel Graph Partitioning and Sparse Matrix Ordering Library Version 4.0. // Technical report, University of Minnesota, Department of Computer Science and Engineering. – 2011. URL: <https://github.com/KarypisLab/ParMETIS/blob/main/manual/manual.pdf>
4. Бартенев Ю.Г., Козинов Е.А., Мееров И.Б., Пирова А.Ю. Решение СЛАУ с разреженной матрицей прямыми методами на суперкомпьютере. Учебное пособие. Н.Новгород: Издательство ННГУ. 2023. 51 с.

Литература

5. Пирова А.Ю. Гибридный MPI + OpenMP алгоритм переупорядочения симметричных разреженных матриц и его применение к решению СЛАУ // Проблемы информатики. – 2022. – №1 (54). – С. 28-41.
6. Pirova A., Meyerov I., Kozinov E., Lebedev S. PMORSy: parallel sparse matrix ordering software for fill-in minimization // Optimization Methods and Software. – 2017. – Vol. 32. – No. 2. – P. 274–289.
7. Пирова А.Ю., Мееров И.Б. Анализ производительности прямого решателя СЛАУ с разреженной матрицей на мини-кластере с процессорами архитектуры RISC-V // Суперкомпьютерные дни в России : Труды международной конференции. 23–24 сентября 2024 г., Москва / Под. ред. Вл. В. Воеводина.. Москва : МАКС Пресс, 2024. – 242 с.. 2024. С. 34-44.

Контакты

Нижегородский государственный университет

<http://www.unn.ru>

Институт информационных технологий, математики и механики

<http://www.itmm.unn.ru>

Кафедра высокопроизводительных вычислений и системного программирования

