



Нижегородский государственный университет им. Н.И. Лобачевского

Факультет Вычислительной математики и кибернетики  
Кафедра Математического обеспечения ЭВМ

# **Оптимизация и применение пакета MUMPS для решения трехмерных стационарных задач прочности на кластерных системах**

Козин Е.А., Мееров И.Б., Сысоев А.В., Бартенев Ю.Г., Бастратов С.И.,  
Лебедев С.А., Лебедев И.Г., Малова А.Ю., Стаканов А.Н.

# Содержание

---

- ❑ Постановка задачи
- ❑ Метод решения
- ❑ Цели работы
- ❑ Выбор MUMPS в качестве базового решателя
- ❑ Тестовая инфраструктура
- ❑ Оптимизация MUMPS для работы с матрицами специального вида
- ❑ Выбор оптимальной конфигурации запуска
- ❑ О решении разреженных СЛАУ большой размерности
- ❑ Заключение



# Постановка задачи

- Пусть дана система линейных уравнений:

$$Ax = b \quad (1)$$

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

- Методы решения:

- Прямые
- Итерационные

- Известные прямые «решатели»: MKL PARDISO, SuperLU, MUMPS, CHOLMOD и др.



# Метод решения...

- Прямой метод решения – разложение Холецкого:

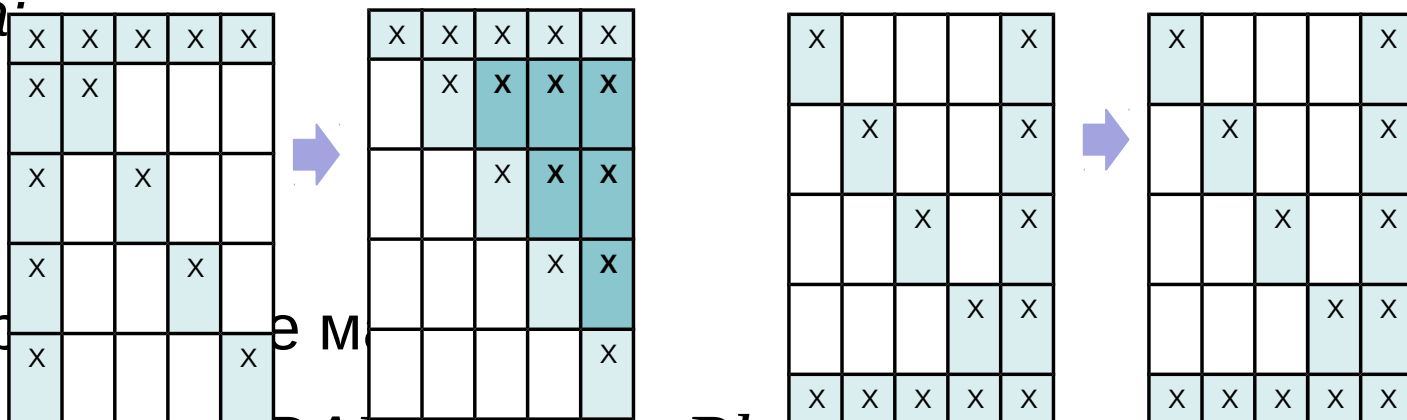
$$A = LL^T \quad (2)$$

$L$  – верхнетреугольная матрица, *фактор* матрицы  $A$

- Переход к решению треугольных систем

$$Lx = b, L^T y = x \quad (3)$$

- Особенность в разреженном случае – *заполнение фактора*:



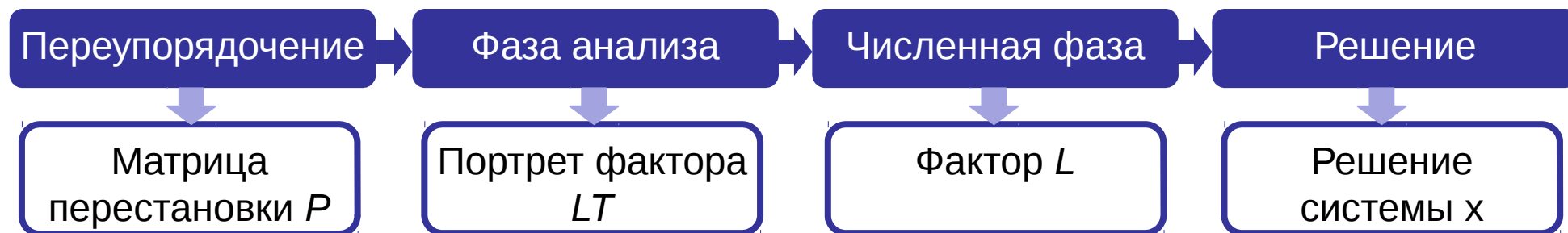
- Переупорядочивание матрицы

$$(PA P^T)(Px) = Pb$$

# Метод решения

- ❑ Методы разложения в разреженном случае:
  - Супернодовый (SuperLU, MKL PARDISO)
  - Мультифронтальный (MUMPS)

- ❑ Вычислительная схема:



# Цели работы

---

- Разработка и реализация параллельной версии прямого метода решения СЛАУ с разреженной SPD-матрицей общего вида для случая LLT разложения, адаптированной для гибридных вычислительных систем с общей и распределенной памятью, оснащенных графическими ускорителями



# Выбор MUMPS в качестве базового решателя

- Основания выбора MUMPS (MULTifrontal Massively Parallel Solver) в качестве базового решателя:
  - Один из ведущих академических прямых решателей разреженных систем вида (1)
  - Регулярно обновляется, реализует обширную функциональность
  - Распространяется бесплатно в исходных кодах по лицензии Public Domain
  - Имеет высокопроизводительную MPI-реализацию, а также комбинированную версию для использования параллелизма на общей и распределенной памяти



# Тестовая инфраструктура

---

- ❑ Кластер ФГУП “РФЯЦ ВНИИЭФ”
  - Использовались от 1 до 36 узлов (12 ядер на каждом)
- ❑ Окружение и сборка
  - ОС Linux
  - Intel C/C++ Compiler, Intel Fortran Compiler из пакета Intel Parallel Studio XE 2011
- ❑ Версии пакетов
  - MUMPS 4.10.0 (BLAS из Intel MKL)
  - переупорядочиватель ParMETIS 4.0
  - Intel MKL PARDISO 2011





# Результаты экспериментов

- Характеристики тестовых матриц. Результаты с использованием Intel MKL PARDISO

Lopatka1	274 104	10 370 664	9,65
Reshetka2	2 263 338	76 684 494	61,82
Trubka	2 428 323	70 338 539	62,08
49_750	2 615 169	96 358 289	Недостаточно памяти*
p4_6	4 216 212	113 040 321	Недостаточно памяти*

\* Intel MKL PARDISO позволяет решить две последние задачи в out-of-core-режиме



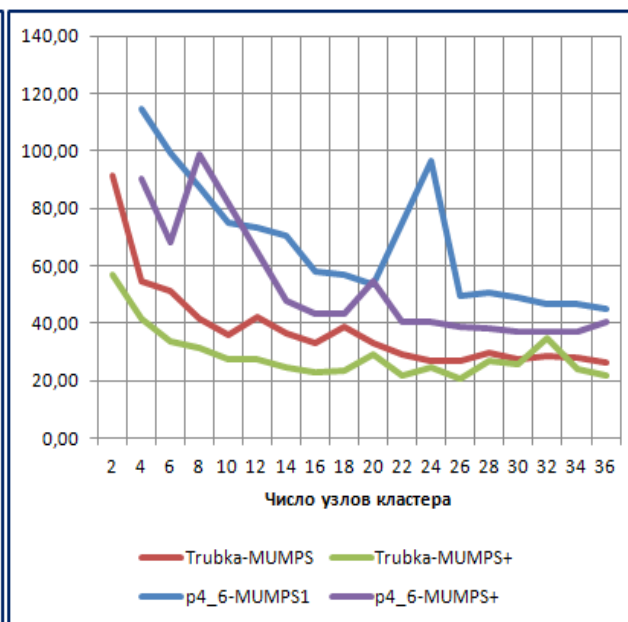
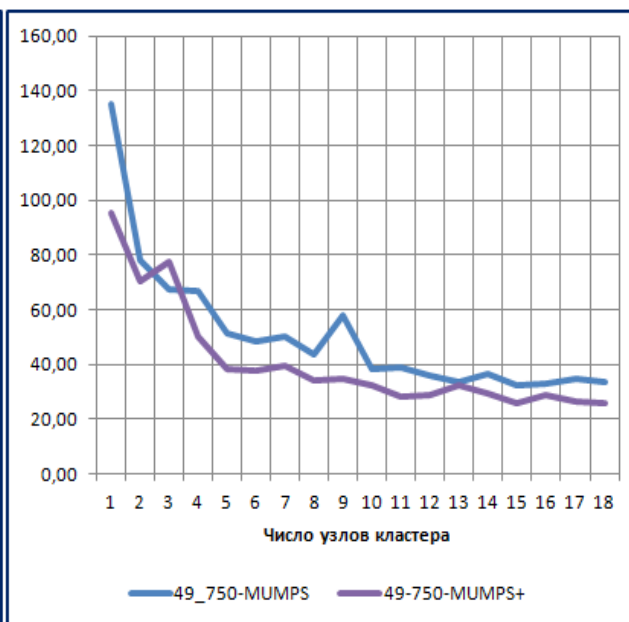
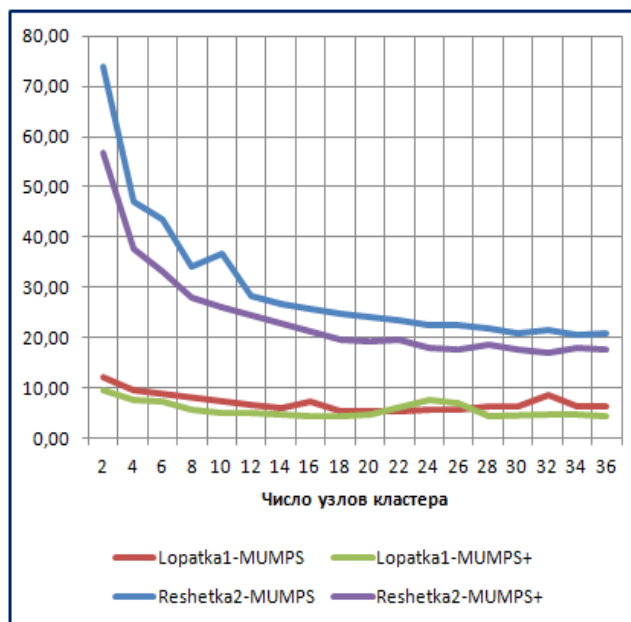
# Оптимизация MUMPS для работы с матрицами специального вида

- ❑ Предпосылки - блочная структура матриц (плотные блоки размером  $3 \times 3$  элемента)
- ❑ Основная идея оптимизации – экономия времени при переупорядочивании матрицы
  - Загрузка портрета матрицы, представленной в блочном формате
  - Переупорядочивания «сжатого» портрета средствами библиотек METIS/ParMETIS
  - «Расширение» полученного результата
- ❑ Результаты
  - Сокращение времени работы стадии переупорядочивания
  - В ряде случаев уменьшение времени численной фазы



# Результаты экспериментов...

- ❑ Оригинальный MUMPS vs MUMPS+ (время в секундах)
  - один MPI-процесс на один узел
  - на каждом узле по 12 потоков



# Результаты экспериментов...

## ❑ Оригинальный MUMPS vs. MUMPS+ (время в секундах)

Lopatka1	5,17 на 20 узлах	4,24 на 28 узлах
Reshetka2	20,65 на 34 узлах	16,97 на 32 узлах
Trubka	26,04 на 36 узлах	20,72 на 26 узлах
49_750	32,58 на 30 узлах	25,66 на 36 узлах
p4_6	45,07 на 36 узлах	36,79 на 30 узлах

Оптимизированная версия позволяет достигать в среднем на 20% лучшего времени решения задач с использованием меньшего числа вычислительных устройств



# Результаты экспериментов...

## □ Intel MKL PARDISO vs. MUMPS+ (время в секундах)

Lopatka1	9,65	9,63
Reshetka2	61,82	56,82
Trubka	62,08	57,08
49_750	Недостаточно памяти	95,34
p4_6	Недостаточно памяти	Недостаточно памяти

При запуске MUMPS+ на 2 узлах удастся улучшить результат Intel MKL PARDISO, запущенного на одном узле



# Результаты экспериментов

- Результаты с использованием Intel MKL PARDISO, MUMPS, MUMPS+ (время в секундах)

Lopatka1	9,65	9,63	5,17 на 20 узлах	4,24 на 28 узлах
Reshetka2	61,82	56,82	20,65 на 34 узлах	16,97 на 32 узлах
Trubka	62,08	57,08	26,04 на 36 узлах	20,72 на 26 узлах
49_750	Недостаточно памяти	95,34	32,58 на 30 узлах	25,66 на 36 узлах
p4_6	Недостаточно памяти	Недостаточно памяти	45,07 на 36 узлах	36,79 на 30 узлах



# Выбор оптимальной конфигурации запуска

- ❑ Варьируемые параметры:
  - число используемых узлов
  - число MPI-процессов на каждый узел
  - число потоков OpenMP в BLAS на каждом процессе
- ❑ Предпосылки:
  - с ростом числа узлов ускорение растет до некоторого предела
  - с ростом числа MPI-процессов на узел растет нагрузка на память
  - с ростом числа потоков на процесс растут затраты на синхронизацию потоков
- ❑ Требуется выбор оптимальной конфигурации



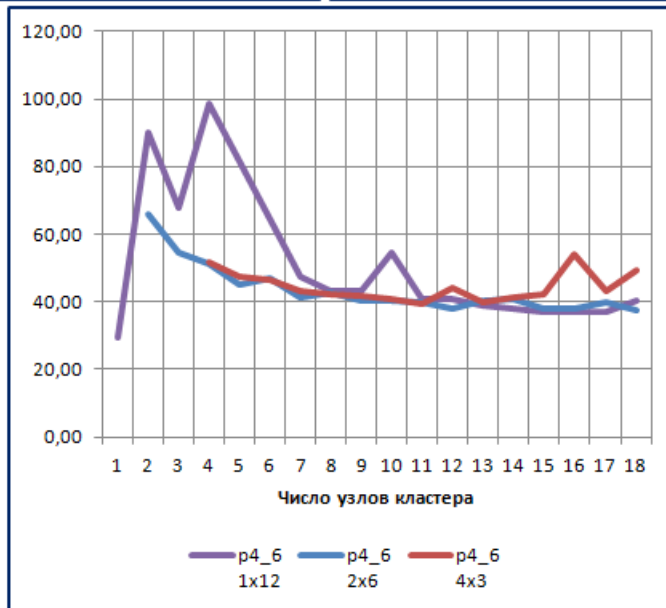
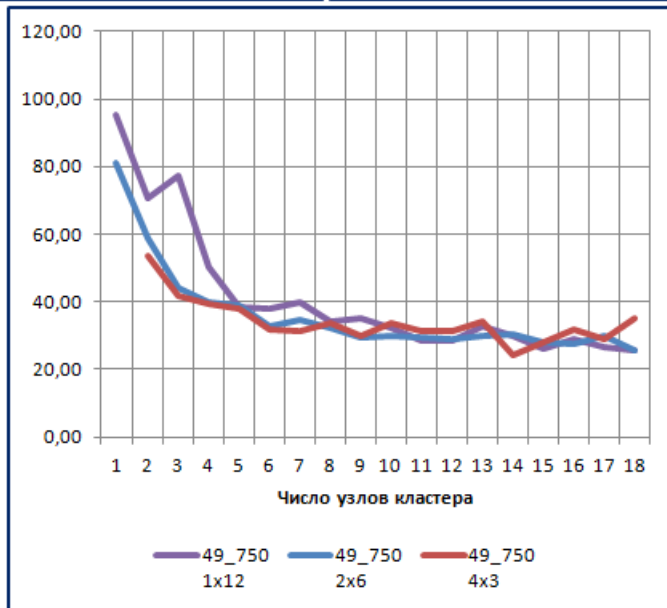
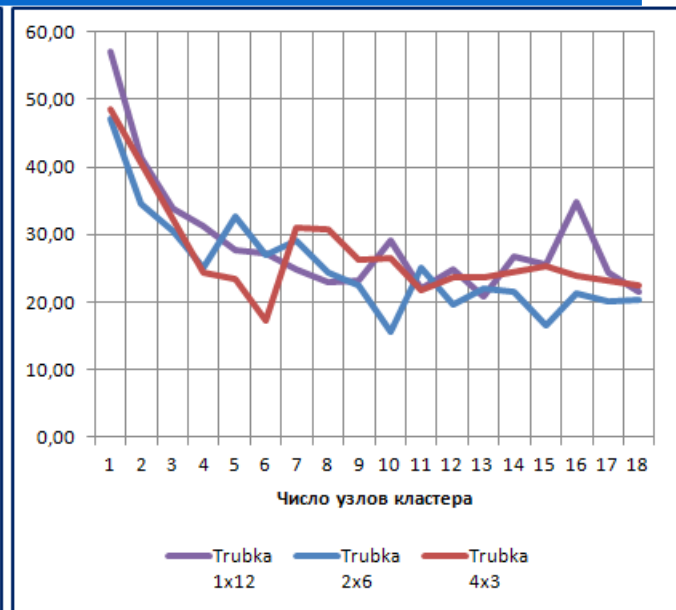
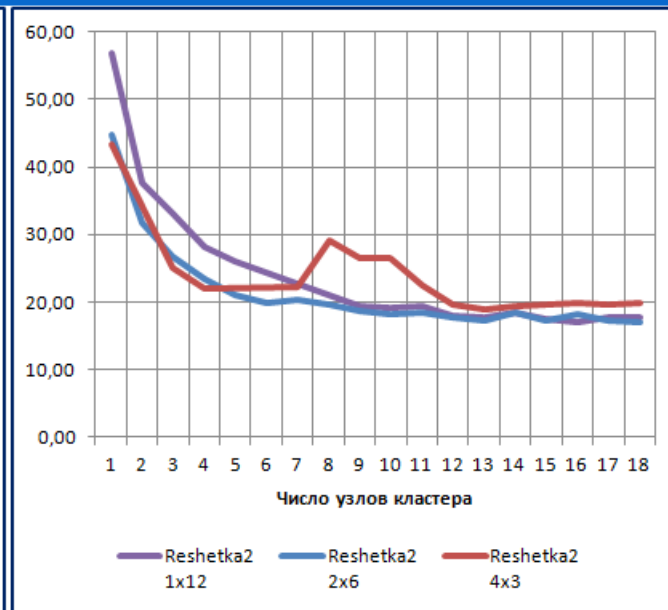
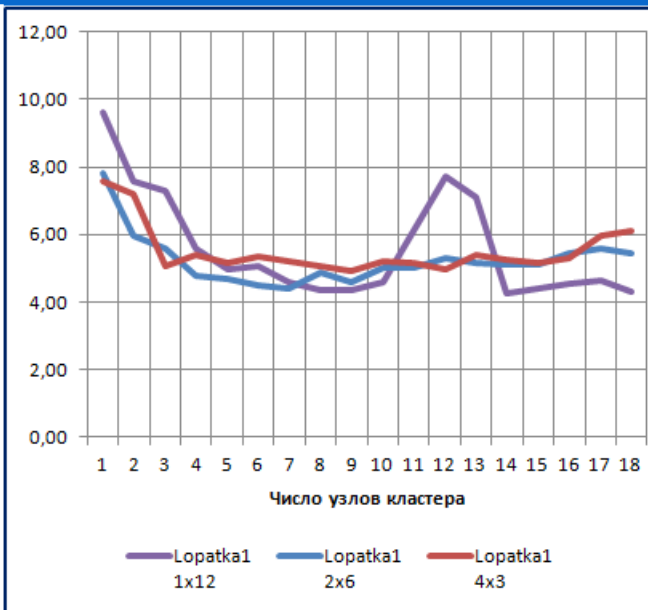
# Результаты экспериментов...

- Пусть используется один узел и
  - $N_p$  – число процессов
  - $N_t$  – число потоков BLAS
  - $N_c$  – число ядер
- Будем выдерживать соотношение  $N_c = N_p \times N_t$
- Сравним 3 режима запуска MUMPS+:
  - $1 \times 12$  (1 MPI-процесс, 12 потоков)
  - $2 \times 6$  (2 MPI-процесса по 6 потоков в каждом)
  - $4 \times 3$  (4 MPI-процесса по 3 потока в каждом)





# Результаты экспериментов



# Анализ результатов экспериментов

- ❑ Наименьшее время показал вариант  $2 \times 6$
- ❑ Изменение режима запуска может значительно (до 2 раз) влиять на время работы
- ❑ Причины:
  - Распараллеливание на потоках в MUMPS выполнено на уровне функций параллельного BLAS третьего уровня.
  - Профилировка показывает, что ключевой операцией является умножение плотных матриц
  - При небольшом размере фронтальных матриц распараллеливание на максимально возможное число потоков на узле может быть не эффективно
- ❑ Необходимо отметить, что с ростом числа используемых узлов доля вычислений на узел сокращается, накладные расходы на передачи по сети увеличиваются, а время работы все меньше зависит от режима запуска



# Оптимизация под графические процессоры

- ❑ В рассматриваемой задаче адаптация под графические процессоры имеет практический смысл только для численной фазы разложения Холецкого
- ❑ Основная идея адаптации – перенос части вычислений BLAS функций на GPU
  - Основное время факторизации – вычисление функций BLAS Level 3
  - Для вычислений возможно использовать высокоэффективные реализации MKL BLAS на CPU и CUBLAS на GPU
  - Адаптация заключена в распределении вычислений между GPU и CPU в зависимости от размера фронтальных матриц
    - CUBLAS эффективен только при больших размерах матриц
    - Низкая эффективность обусловлена большим объемом передаваемых данных



# Результаты экспериментов

- Лучший результат на общей памяти для CPU- и GPU-версий для блочных матриц

Cyclik	1,625	1,656	0,98
Kamaz_Gusev	55,609		1,09
Kamaz_kollekt	1,641	51,063	1,02
Lopatka1	16,844	1,609	1,13
podves	2,906	14,890	0,99
		2,922	

- На относительно маленьких матрицах Cyclik, Kamaz\_kollekt и podves GPU-версия уступает
- На матрицах большего размера Kamaz\_Gusev и Lopatka1 наблюдается ускорение GPU-версии на 8 и 12%, соответственно.



# Заключение

---

- ❑ Выполнена оптимизация решателя MUMPS в последовательной и параллельной версиях для систем с общей и распределенной памятью.
- ❑ Существенный выигрыш по сравнению с последовательной версией решателя достигается за счет использования многопоточного BLAS, использования блочного формата матриц на этапе переупорядочивания и интеграции выполненных модификаций в распределенную версию.



# Вопросы

---

***Спасибо за внимание!***

