

УДК 519.612

ОПТИМИЗАЦИЯ И ПРИМЕНЕНИЕ ПАКЕТА MUMPS ДЛЯ РЕШЕНИЯ ТРЕХМЕРНЫХ СТАЦИОНАРНЫХ ЗАДАЧ ПРОЧНОСТИ НА КЛАСТЕРНЫХ СИСТЕМАХ

С. А. ЛЕБЕДЕВ¹, И. Б. МЕЕРОВ², А. В. СЫСОЕВ³, Ю. Г. БАРТЕНЕВ⁴, С. И. БАСТРАКОВ⁵,
Е. А. КОЗИНОВ⁶, И. Г. ЛЕБЕДЕВ⁷, А. Ю. ПИРОВА⁸, А. Н. СТАКАНОВ⁹

meerov@vmk.unn.ru

^{1-3, 5-8} Нижегородский государственный университет им. Н. И. Лобачевского

^{4, 9} Российский федеральный ядерный центр – Всероссийский научно-исследовательский институт
экспериментальной физики

Поступила в редакцию 31 марта 2014 г.

Аннотация. Предложен подход к оптимизации MUMPS – одного из наиболее известных прямых решателей разреженных СЛАУ, распространяемого в исходных кодах. Основная идея подхода – учет блочной структуры матриц в разреженных СЛАУ, возникающих при решении трехмерных стационарных задач прочности методом конечных элементов. Внедренные на этапе переупорядочивания матрицы улучшения позволили сократить общее время решения задач на ~20% при использовании меньшего числа вычислительных устройств. Рассмотрен вопрос практического применения решателя с учетом конкретного программно-аппаратного окружения. Показана существенная зависимость времени решения задачи от режима запуска (использование разного числа процессов/потоков). Результаты получены на кластере РФЯЦ ВНИИЭФ при решении разреженных СЛАУ с симметричными положительными определенными матрицами.

Ключевые слова: решение СЛАУ; разреженные матрицы; симметричные положительно определенные матрицы; MPI; OpenMP; оптимизация производительности.

ВВЕДЕНИЕ

Одной из актуальных задач алгебры разреженных матриц является решение систем линейных алгебраических уравнений (СЛАУ) $Ax = b$ с разреженной симметричной положительно определенной матрицей A . Такие СЛАУ возникают при решении многих научных и инженерных задач из разных предметных областей. Разработка алгоритмов решения таких задач и их высокопроизводительная программная реализация, ориентированная на современные вычислительные архитектуры, представляет большой практический интерес [1].

На сегодняшний день в мире разработано немало прямых и итерационных решателей СЛАУ, ориентированных на вычислительные системы различной архитектуры (Intel MKL PARDISO, MUMPS, SuperLU, CHOLMOD и другие). Некоторые прямые решатели распространяются в составе инженерных пакетов, таких как, например, ANSYS. Обширные обзоры

прямых решателей можно найти по ссылкам [2, 3]. В последнем обзоре присутствуют только свободно распространяемые решатели, поэтому не упоминается один из наиболее быстрых пакетов для систем с общей памятью – Intel MKL PARDISO [4, 5].

В данной работе рассматривается MUMPS [6] – один из лучших прямых решателей, работающий на кластерных системах традиционной архитектуры, распространяемый в исходных кодах и допускающий использование сторонних высокопроизводительных библиотек. Основное внимание уделяется вопросам оптимизации и выбора режима запуска MUMPS при решении трехмерных стационарных задач прочности, формируемых пакетом программ «ЛОГОС Прочность» [7]. Работа выполнена по заказу ФГУП «РФЯЦ-ВНИИЭФ».

Статья построена следующим образом: приведена краткая постановка задачи решения СЛАУ, обоснован выбор MUMPS в качестве базового решателя, сформулированы основные

идеи по оптимизации решателя с учетом блочной структуры матриц, показана важность выбора подходящей конфигурации запуска на кластере, приведены результаты вычислительных экспериментов и их анализ, в заключении обсуждены перспективы адаптации MUMPS для ускорителей GPU и Intel Xeon Phi.

ПОСТАНОВКА ЗАДАЧИ И МЕТОД РЕШЕНИЯ

Пусть дана система линейных уравнений $Ax = b$, где A – разреженная симметричная положительно определенная матрица, b – плотный вектор, x – искомый вектор неизвестных. Прямые методы решения системы, как правило, основаны на применении разложения Холецкого к матрице A с последующим решением двух треугольных систем. При выполнении разложения Холецкого матрица обычно претерпевает заполнение, что на практике может привести к неудовлетворительным требованиям по памяти. Степень заполненности матрицы можно уменьшить с помощью переупорядочивания ее строк и столбцов. При решении системы с использованием разложения Холецкого можно выделить следующие этапы: переупорядочивание, символическое разложение (построение портрета фактора и выделение памяти для хранения ненулевых элементов), численное разложение (вычисление фактора) и решение треугольных систем. Численное разложение может выполняться одним из трех методов: ориентированным влево, ориентированным вправо, мультифронтальным. Подробное описание постановки задачи и мультифронтального метода приведено в работе [8].

ВЫБОР MUMPS В КАЧЕСТВЕ БАЗОВОГО РЕШАТЕЛЯ

MUMPS (MUltifrontal Massively Parallel Solver) – программный комплекс для решения разреженных СЛАУ [9]. Данный комплекс позволяет решать системы как с симметричными, так и с несимметричными матрицами. Причины выбора MUMPS в качестве базового решателя состоят в следующем:

1. MUMPS является одним из ведущих академических прямых решателей разреженных СЛАУ [6]. Он разрабатывается с 1996 г. в университетах Лиона, Тулузы, Бордо, регулярно обновляется, реализует обширную функциональность, имеет хорошую репутацию и большой спектр приложений [6].

2. Текущая версия MUMPS 4.10.0 распространяется бесплатно в исходных кодах по лицензии Public Domain, что допускает его пере-

работку сторонними группами и не накладывает существенных ограничений на использование.

3. MUMPS имеет высокопроизводительную MPI-реализацию для распределенных систем, дополнительные возможности по работе с большими матрицами (использование режима out-of-core), а также комбинированную версию для использования параллелизма на общей и распределенной памяти. Все это позволяет решателю показывать хорошие результаты производительности в сравнении с другими академическими и коммерческими решателями.

ОПТИМИЗАЦИЯ MUMPS ДЛЯ РАБОТЫ С МАТРИЦАМИ СПЕЦИАЛЬНОГО ВИДА

Разреженные матрицы, возникающие в процессе решения трехмерных стационарных задач прочности методом конечных элементов, имеют нерегулярную разреженную структуру из плотных блоков размером 3×3 элемента. Такую матрицу размера $N \times N$ можно рассматривать как матрицу размера $K \times K$, при $K = N/3$, где элементами являются плотные блоки размером 3×3 . Основная идея выполненной оптимизации заключается в экономии времени при переупорядочивании матрицы с целью сокращения заполнения фактора. Экономия достигается за счет загрузки портрета матрицы, представленной в блочном формате, и его переупорядочивания средствами библиотек METIS/ParMETIS [10, 11] с последующим «расширением» полученного результата. При этом время работы стадии переупорядочивания сокращается (так, для матриц, состоящих из плотных блоков размера 3×3 , число вершин в графе уменьшается в 3 раза). Кроме того, в ряде случаев может уменьшаться время работы на следующих стадиях алгоритма в связи с тем, что эвристический алгоритм поиска перестановки потенциально может найти лучшее решение в задаче меньшей размерности.

ВЫБОР ОПТИМАЛЬНОЙ КОНФИГУРАЦИИ ЗАПУСКА

Выбор подходящей конфигурации запуска прикладного ПО, ориентированного на кластерные системы, часто весьма непросто. Так, например, при решении задачи нахождения собственных значений и векторов разреженной матрицы (в частности, при помощи известного пакета SLEPc) для каждой конкретной задачи требуется «угадать» достаточное количество используемых ядер, а также параметр, регулирующий размер создаваемых вспомогательных структур данных. Неудачное сочетание этих параметров

может привести к аварийному завершению задачи в связи с нехваткой памяти, в связи с истечением лимита времени на одну задачу на кластере (либо неудовлетворительному времени счета при отсутствии лимита) и другим нежелательным последствиям. При решении разреженных СЛАУ прямыми методами неудачный выбор числа используемых процессоров обычно порождает аналогичные проблемы. Так, в некоторых случаях объемы требуемой оперативной памяти на один узел могут превышать объем доступной установленной памяти, что вызывает аварийный останов или замедление работы на порядки в связи с использованием файла подкачки. В связи с указанными причинами задача выбора подходящей конфигурации запуска является достаточно актуальной.

В некоторых математических библиотеках и пакетах численного моделирования решению указанной проблемы уделяется значительное внимание. Так, например, в библиотеке ATLAS при сборке и установке выполняется калибровка параметров и настройка под конкретную вычислительную систему. Аналогичный подход применяется в пакете алгоритмов линейной алгебры MAGMA, ориентированном на работу с плотными матрицами на гетерогенных архитектурах (CPU + GPU, CPU + Xeon Phi).

Запуск специально подготовленных тестов для настройки под конкретное программно-аппаратное окружение не является единственным способом выбора параметров алгоритмов. Так, в некоторых библиотеках (Intel MKL, GOTO BLAS и др.) подобная настройка, судя по всему, выполняется при помощи определения параметров конкретной вычислительной системы (поддерживаемые наборы команд, размер кеш-памяти, длина кеш-линейки и др.) непосредственно при запуске. При этом можно применять разные подходы, а именно: параметризация программного кода, разработка разных реализаций для одной и той же функции с использованием разных наборов команд.

К сожалению, в обсуждаемых в данной работе алгоритмах решения разреженных СЛАУ выбор оптимального режима запуска затруднен.

Конечно, настройка параметров основных математических ядер, в частности, умножения плотных матриц, выполняется аналогичными способами, но оценка размеров необходимой памяти и тем более оптимального соотношения числа процессов/потоков на одном узле является достаточно сложной задачей. Необходимо отметить, что MUMPS успешно справляется с оценкой необходимых объемов памяти без помощи пользователя. Тем не менее выбор оптимального числа процессов MPI и потоков параллельного BLAS на вычислительном узле необходимо выполнять самостоятельно. В данной работе приводятся результаты экспериментов с распределенной версией MUMPS при варьировании числа процессов MPI/потоков OpenMP в BLAS, обсуждается возможность выбора подходящей конфигурации запуска.

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для анализа производительности программного комплекса были проведены эксперименты на кластере ФГУП «РФЯЦ ВНИИЭФ», с использованием 1–36 узлов, 12 ядер на каждом узле. Характеристики тестовых матриц представлены в таблице ниже (табл. 1, столбцы 1–4). Эти СЛАУ получены из трехмерных стационарных задач прочности, выработанных пакетом программ «ЛОГОС Прочность» [7]. Все они являются разреженными, симметричными положительно определенными и имеют описанную ранее блочную структуру. Использовались ОС Linux, Intel MKL PARDISO, Intel C/C++ Compiler, Intel Fortran Compiler из пакета Intel Parallel Studio XE 2011 и MUMPS 4.10.0. При сборке решателей использовался BLAS из Intel MKL, переупорядочиватель ParMETIS 4.0.

Результаты с использованием MKL PARDISO. Для получения базы для дальнейшего анализа результатов использовался решатель Intel MKL PARDISO из пакета Intel Parallel Studio XE 2011 в in-core-режиме для решения указанных задач. Intel MKL PARDISO не поддерживает распределенный режим работы, но демонстрирует отличные результаты на одном узле кластера. Время работы решателя приведено в табл. 1 (столбец 4).

Таблица 1

Характеристики тестовых матриц. Результаты с использованием Intel MKL PARDISO

Название матрицы	Порядок	Число ненулевых элементов	Время решения с использованием 12 ядер, секунды
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 с использованием жесткого диска в качестве «дополнительной памяти», однако сравнение результатов Intel MKL PARDISO в out-of-core-режиме и MUMPS в in-core-режиме (за счет большего объема памяти на нескольких узлах кластера) не имеет смысла.

Результаты оптимизации MUMPS для работы с матрицами специального вида. Рассмотрим результаты экспериментов для оригинальной версии MUMPS и версии MUMPS+, оптимизированной для работы с блочными матрицами (рис. 1). На рисунке отображено время работы MUMPS и MUMPS+ при изменении числа используемых узлов от 2 до 36. По оси ординат отложено время в секундах, по оси абсцисс – число узлов кластера. Обе версии MUMPS запускались в режиме один MPI-процесс на один узел. На каждом узле создавалось 12 потоков (по числу вычислительных ядер).

Из графиков видно, что при увеличении числа используемых узлов кластера время решения задачи ожидаемо уменьшается. Тем не менее, в некоторый момент наступает «насыщение», время стабилизируется и даже может начать расти. Этот эффект особенно заметен на матрице Lopatka1, что, впрочем, тоже ожидаемо: эта матрица имеет сравнительно небольшие размер и число ненулевых элементов. Начиная с некоторого момента накладные расходы на поддержку параллельного режима работы становятся сопоставимы с выигрышем, получаемым от увеличения числа вычислительных ядер.

В табл. 2 представлено наименьшее время решения задачи с использованием MUMPS и MUMPS+ с указанием числа узлов, на котором оно достигается, а также время MUMPS+ на двух узлах и для сравнения время Intel MKL

PARDISO на одном узле.

Результаты показывают, что при запуске MUMPS+ на двух узлах удается улучшить результат Intel MKL PARDISO, запущенного на одном узле, что говорит о применимости оптимизированной версии MUMPS в сравнении с одним из лучших коммерческих решателей SLAU. Однако необходимо отметить, что описанный выше подход к оптимизации вычислений можно применить в любом прямом решателе, что должно привести к улучшению результатов при решении задач с блочной структурой.

Сравнение базовой и оптимизированной версий MUMPS показывает, что оптимизированная версия позволяет достигать в среднем на 20% лучшего времени решения задач с использованием меньшего числа вычислительных устройств.

Выбор оптимальной конфигурации запуска. Рассмотрим результаты работы по дальнейшему уменьшению времени счета при помощи выбора оптимального числа процессов/потоков на одном узле. Пусть Np – число процессов, Nt – число потоков BLAS, а Nc – число ядер. Будем выдерживать соотношение $Nc = Np \times Nt$. Сравним три режима запуска оптимизированной версии MUMPS+: 1×12, 2×6, 4×3 (рис. 2).

Наименьшее время показал вариант 2×6, при этом изменение режима запуска может значительно (до двух раз) влиять на время работы, особенно при небольшом числе узлов. Это объясняется тем, что распараллеливание на потоках в MUMPS выполнено на уровне функций параллельного BLAS третьего уровня. Профилировка показывает, что ключевой операцией при этом является умножение плотных матриц, которое далеко не в каждом вызове эффективно распараллеливается на максимально возможное число потоков на узле.

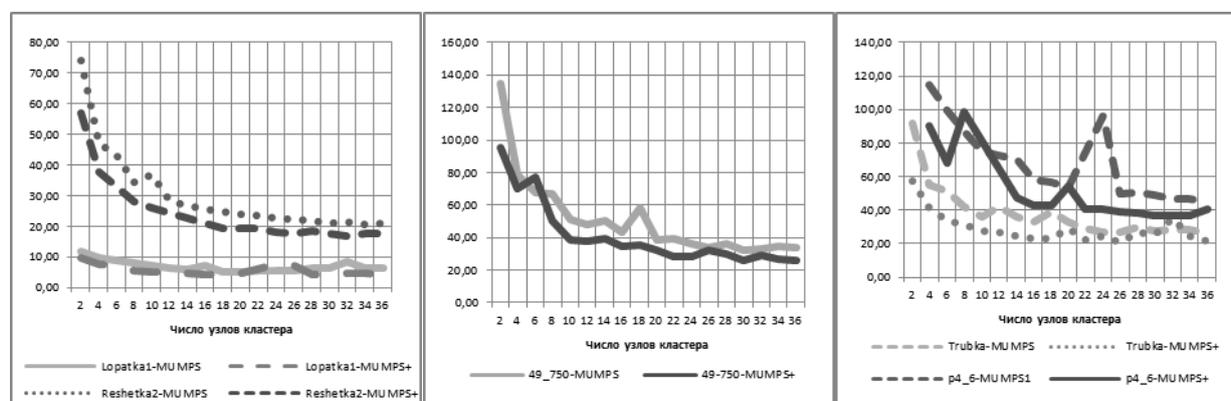


Рис. 1. Сравнение базовой и оптимизированной версий MUMPS

Результаты с использованием Intel MKL PARDISO, MUMPS, MUMPS+

Название матрицы	Время решения, PARDISO, секунды	Время MUMPS+ на 2 узлах, секунды	Лучшее время, MUMPS, секунды	Лучшее время, MUMPS+, секунды
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

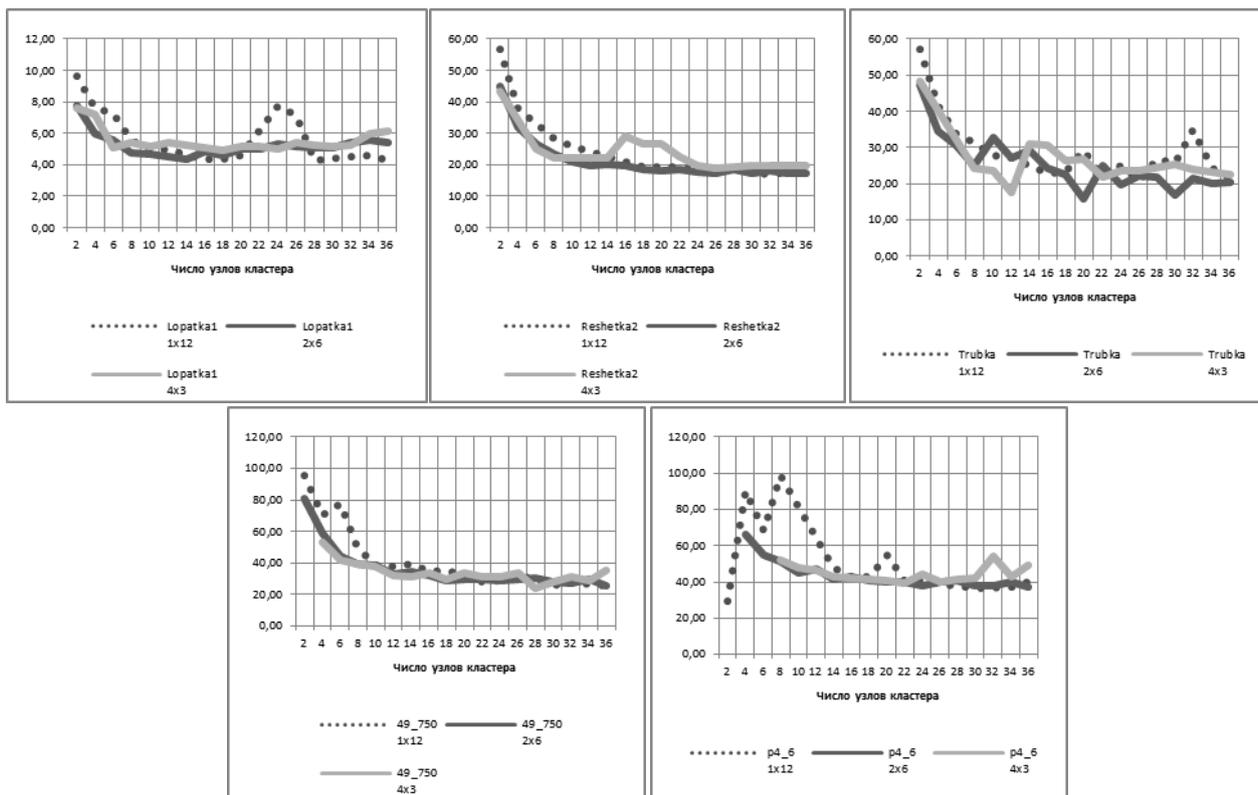


Рис. 2. Выбор оптимальной конфигурации запуска

В таких случаях использование параллелизма на более высоком уровне может привести к сокращению общего времени счета. 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.

ЗАКЛЮЧЕНИЕ

Обзор текущего положения дел в области решения СЛАУ с разреженной симметричной положительно определенной матрицей прямыми методами показывает наличие значительного числа программных комплексов, ориентирован-

ных как на системы с общей памятью, так и на кластерные системы, распространяемых как отдельно, так и в составе проблемно-ориентированных инженерных пакетов. Среди таких программных комплексов – один из лучших свободно распространяемых решателей с открытым кодом MUMPS, ориентированный на кластерные системы, и известный коммерческий решатель Intel MKL PARDISO, оптимизированный для многоядерных систем. В статье рассказывается об опыте оптимизации пакета MUMPS для решения больших разреженных СЛАУ, возникающих при решении трехмерных стационарных задач прочности методом конечных элементов. Суть оптимизации заключается

в учете блочной структуры задач на этапе перепорядочивания матрицы. Эксперименты показывают сокращение общего времени счета на ~ 20 %. В работе изучается влияние режима запуска (число узлов кластера, соотношение числа MPI-процессов и потоков BLAS на одном узле) на время счета, демонстрируется существенное (до 2 раз при небольшом числе узлов) изменение времени счета в зависимости от конфигурации запуска. Оптимизированный решатель MUMPS подключен Заказчиком к комплексу библиотек параллельных решателей SLAU LParSol [12].

Направления дальнейшей деятельности могут быть связаны с переносом части вычислений на ускорители GPU и Intel Xeon Phi. Вопрос о возможности эффективной адаптации для ускорителей прямых решателей разреженных SLAU [13] изучается достаточно давно, но все еще является открытым. Один из наиболее простых способов адаптации в рассматриваемой задаче состоит в переносе функций BLAS на GPU или Xeon Phi. При этом возможно использование существующих высокопроизводительных библиотек (CUBLAS для GPU, MKL для Xeon Phi, MAGMA для гетерогенных систем). Указанный способ требует специального алгоритма выбора устройства для выполнения функций в зависимости от размера задачи. Большинство реализаций BLAS для ускорителей эффективны только для матриц достаточно большого размера. Очевидно, в большинстве случаев данный подход не приведет к ускорению вычислений в связи с малыми размерами матриц, обрабатываемых функциями BLAS. Исследование возможности создания эффективных реализаций рассмотренных алгоритмов для ускорителей является направлением дальнейшей работы.

СПИСОК ЛИТЕРАТУРЫ

1. **Козинев Е. А.**, Лебедев И. Г., Лебедев С. А., Малова А. Ю., Мееров И. Б., Сысоев А. В., Филиппенко С. С. Новый решатель для алгебраических систем разреженных линейных уравнений с симметричной положительно определенной матрицей // Вестник Нижегородского университета им. Н. И. Лобачевского. 2012. № 5 (2). С. 376–384. [Е. А. Kozinov, I. G. Lebedev, S. A. Lebedev, et al., "Novel linear equations systems solver for sparse symmetric positive-definite matrix," (in Russian), in *Vestnik UNN*, no. 5 (2), pp. 376–384, 2012.]
2. **Li X.** Direct Solvers for Sparse Matrices. [Электронный ресурс]. URL: <http://crd-legacy.lbl.gov/~xiaoye/SuperLU/SparseDirectSurvey.pdf>. [X. Li. *Direct Solvers for Sparse Matrices* [Online]. Available: <http://crd-legacy.lbl.gov/~xiaoye/SuperLU/SparseDirectSurvey.pdf>]
3. **Dongarra J.** Freely Available Software for Linear Algebra [Электронный ресурс]. URL: <http://www.netlib.org/utk/people/JackDongarra/la-sw.html>. [J. Dongarra. *Freely Available Software for Linear Algebra* [Online]. Available: <http://www.netlib.org/utk/people/JackDongarra/la-sw.html>]
4. **Intel Math Kernel Library Reference Manual** [Электронный ресурс]. URL: <http://software.intel.com/sites/products/documentation/hpc/mkl/mklman.pdf>. [Intel Math Kernel Library Reference Manual [Online]. Available: <http://software.intel.com/sites/products/documentation/hpc/mkl/mklman.pdf>]
5. **Intel MKL web page** [Электронный ресурс]. URL: <http://software.intel.com/en-us/intel-mkl>. [Intel MKL web page [Online]. Available: <http://software.intel.com/en-us/intel-mkl>]
6. **MUMPS web page** [Электронный ресурс]. URL: <http://gral.ens-lyon.fr/MUMPS>. [MUMPS web page [Online]. Available: <http://gral.ens-lyon.fr/MUMPS>]
7. **Речкин В. Н.**, Спиридонов В. Ф., Цибарев К. В., Дьянов Д. Ю., Наумов А. О., Косарим С. С., Филимонкин Е. А., Бартнев Ю. Г., Щаникова Е. Б., Ерзунов В. А., Рябов В. А., Вяткин Ю. А. Пакет программ ЛОГОС. Модуль решения квазистатических задач прочности и модального анализа // Супервычисления и математическое моделирование: тр. XIII Междунар. сем. Саров, 2011. [V.N. Rechkin, et al., "Software package LOGOS. Modul for solving the quasi-static tasks of strength and modal analysis," in *Proc. XIII International seminar "Supercomputations and mathematical modeling"*, Sarov, 2011.]
8. **Liu J.** The multifrontal method for sparse matrix solution: Theory and practice // SIAM Review. 1992. 34. P. 82–109. [J. Liu, "The multifrontal method for sparse matrix solution," *SIAM Review*, no. 34, pp. 82–109, 1992.]
9. **Multifrontal Massively Parallel Solver (MUMPS 4.10.0)** User's guide. Technical report. ENSEEINT-IRIT, 2011 [Электронный ресурс]. URL: http://mumps.enseeiht.fr/doc/userguide_4.10.0.pdf. [Multifrontal Massively Parallel Solver (MUMPS 4.10.0) User's guide. Technical report. ENSEEINT-IRIT, 2011 [Online]. Available: http://mumps.enseeiht.fr/doc/userguide_4.10.0.pdf]
10. **Karipis G.** METIS. A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Version 5.0. Technical report [Электронный ресурс]. University of Minnesota, Department of Computer Science and Engineering, 2011. URL: <http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/manual.pdf>. [G. Karipis, METIS. *A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Version 5.0*. Technical report [Online]. Available:]
11. **Karypis G.**, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs // SIAM J. on Scientific Computing. 1999. Vol. 20, No. 1. P. 359–392. [G. Karypis, V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. on Scientific Computing*, Vol. 20, No. 1. pp. 359–392, 1999.]
12. **Бартнев Ю. Г.**, Бондаренко Ю. А., Ерзунов В. А. и др. Комплекс LParSol для решения SLAU // Супервычисления и математическое моделирование: сб. тез. докл. XIII Междунар. сем. Саров, 2011. С. 34–36. [Yu.G. Bartnev, Yu.A. Bondarenko, V.A. Erzunov, et al., "Software package LParSol for SLAE solving," in *Proc. XIII International seminar "Supercomputations and mathematical modeling"*, Sarov, pp. 34–36, 2011.]
13. **Posey S.** GPU progress in sparse matrix solvers for applications in computational mechanics // European seminar on

computing (Pilzen, Czech Republic, June, 2012). [S. Posey, "GPU progress in sparse matrix solvers for applications in computational mechanics," in *Proc. European seminar on computing* (Pilzen, Czech Republic, June, 2012).]

ОБ АВТОРАХ

ЛЕБЕДЕВ Сергей Александрович, магистрант каф. мат. обеспечения ЭВМ (ВМК, ННГУ). Б-р прикл. мат. и инф. (ННГУ, 2012). Иссл. в обл. разработки высокопроизводительных решателей разреженных СЛАУ.

МЕЕРОВ Иосиф Борисович, доц. каф. мат. обеспечения ЭВМ (ВМК, ННГУ). М-р в обл. прикл. мат. и информатики (ННГУ, 1999). Канд. техн. наук по мат. моделированию, числ. методам и комплексам программ (ННГУ, 2005), доцент. Иссл. в обл. исп. суперкомпьютеров для решения прикладных задач.

СЫСОЕВ Александр Владимирович, асс. каф. мат. обеспечения ЭВМ (ВМК, ННГУ). М-р в обл. прикл. мат. и информатики (ННГУ, 1999). Канд. техн. наук по мат. моделированию, числ. методам и комплексам программ (ННГУ, 2012). Иссл. в обл. параллельного программирования и глобальной оптимизации.

БАРТЕНЕВ Юрий Германович, гл. науч. сотр., ВНИИЭФ. Дипл. механик (1972, МГУ им. М. В. Ломоносова). Д-р физ.-мат. наук по выч. машинам, комплексам и сетям (ВНИИЭФ, 1998). Иссл. в обл. параллельных алгоритмов.

БАСТРАКОВ Сергей Иванович, асс. каф. мат. обеспечения ЭВМ (ВМК, ННГУ). М-р в обл. прикл. мат. и информатики (ННГУ, 2012). Иссл. в обл. использования гетерогенных вычислительных систем.

КОЗИНОВ Евгений Александрович, асс. каф. мат. обеспечения ЭВМ (ВМК, ННГУ). М-р в обл. прикл. мат. и информатики (ННГУ, 2007). Иссл. в обл. разработки визуальных сред для параллельного программирования.

ЛЕБЕДЕВ Илья Геннадьевич, программист каф. мат. обеспечения ЭВМ (ВМК, ННГУ). М-р в обл. прикл. мат. и информатики (ННГУ, 2012).

ПИРОВА (Малова) Анна Юрьевна, асп. каф. мат. обеспечения ЭВМ (ВМК, ННГУ). М-р в обл. прикл. мат. и информатики (ННГУ, 2013). Иссл. в области методов переупорядочения разреженных матриц.

СТАКАНОВ Артем Николаевич, науч. сотр.. Дипл. математик-программист (СПбГУ, 2005). Иссл. в обл. разработки мат. библиотек.

METADATA

Title: Performance optimization and application of the MUMPS for solving 3-dimensional stationary strength problems on cluster systems.

Authors: S. A. Lebedev¹, I. B. Meyerov¹, A. V. Sysoyev¹, Yu. G. Bartenev², S. I. Bastrakov¹, E. A. Kozinov¹, I. G. Lebedev¹, A. Yu. Pirova¹, A. N. Stakanov².

Affiliation:

¹ Lobachevsky State University of Nizhni Novgorod (UNN), Russia.

² Russian Federal Nuclear Center. All-Russian Research Institute Of Experimental Physics (FSUE RFNC – VNIIEF), Russia.

Email: meerov@vmk.unn.ru.

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 18, no. 3 (64), pp. 276-282, 2014. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: We present an approach to performance optimization of MUMPS, which is one of the most widely used famous open source sparse SLAE solvers. The main idea is to take into account special block structure of sparse matrixes arising from the finite element analysis of 3-dimensional stationary strength problems. Implemented in the reordering phase, this approach yields about 20 % reduction of total computational time while using fewer cores. Next, we consider application and adjustment of the solver for a specific machine. We demonstrate that the threads/processes configuration significantly influences performance. The results are obtained on the FSUE RFNC – VNIIEF cluster for symmetric positive defined matrixes.

Key words: SLAE solution; sparse matrixes; symmetric positive defined matrixes; MPI; OpenMP; performance optimization.

About authors:

LEBEDEV, Sergey Alexandrovich, Master's degree student of Software Department (CMC, UNN). Research in high-performance solvers of sparse SLAE.

MEYEROV, Iosif Borisovich, PhD., associate professor of Software Department (CMC, UNN). Research in applications of supercomputers.

SYSOYEV, Alexander Vladimirovich, PhD., assistant professor of Software Department (CMC, UNN). Research in parallel programming and global optimization.

BARTENEV, Yury Germanovich, principal scientific worker of RFNC-VNIIEF, research in math. libraries and software.

BASTRAKOV, Sergey Ivanovich, assistant teacher of Software Department (CMC, UNN). Research in heterogeneous computing.

KOZINOV, Evgeny Alexandrovich, assistant teacher of Software Department (CMC, UNN). Research in visual environments for parallel programming.

LEBEDEV, Ilya Genadievich, software engineer of Software Department (CMC, UNN).

PIROVA (MALOVA), Anna Yurievna, postgraduate student of Software Department (CMC, UNN). Research in reordering of sparse matrixes.

STAKANOV, Artem Nikolaevich, scientific worker of RFNC-VNIIEF, research in math. libraries.