

## КОМПЛЕКС БИБЛИОТЕК ПАРАЛЛЕЛЬНЫХ РЕШАТЕЛЕЙ СЛАУ LPARSOL ВЕРСИИ 3

*Ю. Г. Бартенева<sup>1,2</sup>, В. А. Ерзунов<sup>1</sup>, А. П. Карпов<sup>1</sup>, Д. А. Петров<sup>1</sup>, И. А. Пищулин<sup>1</sup>,  
А. Н. Стаканов<sup>1</sup>, Е. Б. Щаникова<sup>1</sup>, И. Е. Капорин<sup>3</sup>, О. Ю. Милюкова<sup>4</sup>,  
С. А. Харченко<sup>5</sup>, И. Н. Коньшин<sup>5</sup>, И. Б. Мееров<sup>2</sup>, А. В. Сусоев<sup>2</sup>*

<sup>1</sup>Российский федеральный ядерный центр –  
Всероссийский НИИ экспериментальной физики, г. Саров,  
<sup>2</sup>Нижегородский государственный университет им. Н. И. Лобачевского,  
г. Нижний Новгород,  
<sup>3</sup>Вычислительный центр им. А. А. Дородницына РАН, г. Москва,  
<sup>4</sup>Институт прикладной математики им. М. В. Келдыша РАН, г. Москва,  
<sup>5</sup>ОАО ТЕСИС, г. Москва

В отчете [1] американских специалистов обращено внимание на важность:

- продолжения инвестиций в разработку библиотек (PETSc, HYPRE, Trilinos, SuperLU и др.), благодаря чему многие решатели стали общедоступны;
- развития многоуровневых (многосеточных) алгоритмов, так как они математически оптимальны, высокопараллельны, часто – наискорейшие и широко применяемы;
- развития прямых и итерационных решателей, использующих коммуникационно ненапряженные (communication-avoiding) алгоритмы и другие, вызванные ограничениями архитектуры компьютеров и перекликается с исследованиями в России [2–5].

В 2010–2013 гг. по проекту суперкомпьютерных технологий ряд предприятий совместно разработали библиотеку решателей СЛАУ в виде комплекса библиотек LParSol [6], который включает библиотеки PMLP/ParSol, TParFBSS, VC\_RAN\_SLAU [6, 2] с единым пользовательским интерфейсом и средства подключения библиотек Intel PARDISO MKL [7], предобусловливателя BoomerAMG HYPRE [8], MUMPS [9], предобусловливателя AMG Parallel Toolbox [10]. Прямой симметричный решатель MUMPS адаптирован к обработке мелкоблочных матриц [11], AMG Parallel Toolbox доработан и адаптирован к потоку СЛАУ [12].

Объединяющая оболочка LParSol обеспечивает решение последовательности распределенных СЛАУ, отличающейся структуры, сложности на одном и разных коммутаторах MPI. Ниже перечислены характеристики и эффект:

- формирование СЛАУ не зависит от решателя => решать можно любым;
- внешнее задание решателя и его настроек => менять, не меняя код;
- ориентация на поток СЛАУ => снижает затраты, адаптивность;
- разнотипные потоки СЛАУ в одной задаче => для каждого свой решатель;
- блочные форматы матриц => экономия памяти, путь к векторизации;
- mixed-precision алгоритмы=> экономия памяти, ускорение вычислений;
- итерационные решатели => предобусловленные в подпространствах Крылова;
- предобусловливатели => Schwartz (ILU/IC), сквозные (ILU), упрощенный AMG;
- прямые решатели => небольшие задачи, Schwartz (LU/LDLt);
- применение => во многих кодах ВНИИЭФ.

В LParSol для распределенных СЛАУ реализованы:

- итерационные решатели CG, BiCGStab, CR, BiCRStab, GMRES, SOFGMRES;
- сквозной предобусловливатель IC2/ILU2 [13];

- блочный предобусловливатель (ADI, mcILU0, IC0/ILU0, ICk/ILUk, ICt/ILUt);
- блочно-перекрытый предобусловливатель (ICt/ILUt, IC2/ILU2/MILUt, LLt/LU);
- крупноблочно-перекрытый предобусловливатель (IC2/ILU2);
- грубосеточный корректировщик поправки предобусловливания;
- оптимизированный селективный AMG [10, 12];
- оптимизированный для блочных матриц LDLt решатель MUMPS.

Особенностями LParSol являются:

- единый формат СЛАУ и задание метода и параметров решения в файле (можно менять решатель, не изменяя приложение);
- одинарная и двойная точность предобусловливания (IC/ILU, IC2/ILU2) [3, 6];
- точечный и блочный разреженный формат матриц в ряде решателей [3, 12];
- оптимизирующие переупорядочивания матрицы (RCM, ND, др.);
- наследование построенных структур решателя предыдущей СЛАУ;
- адаптивный выбор для потока СЛАУ наискорейшего решателя [14].

Ряд методов, демонстрирующих либо лучшее качество предобусловливания, либо более высокий параллелизм, либо то и другое: многоцветные ILU, прямой мультипоточный LLt решатель, полиномиальный предобусловливатель [2], еще находится в стадии испытаний.

Дополнения третьей версии LParSol включают параллельное:

- предобусловливание методом Шварца (исходно в библиотеке ВЦ РАН [15], затем с некоторыми расширениями в TParFBSS и PMLP/ParSol) на основе IC2, MLILU, ILU2, IC, ILU, LL<sup>T</sup>, LU в перекрытых подобластях (представлен отдельным докладом);
- предобусловливание селективным AMG (Parallel Toolbox с доработками);
- уточнение предобусловливания методом Coarse Grid Correction [16] (PMLP/ParSol);
- решение СЛАУ на ЭВМ с распределенной памятью прямым методом (MUMPS, усовершенствованный в части LDLt-решателя для мелкоблочных матриц).

Ряд методов, демонстрирующих либо лучшее качество предобусловливания, либо более высокий параллелизм, либо то и другое: многоцветные ILU, прямой мультипоточный LLt решатель, полиномиальный предобусловливатель [2], еще находится в стадии испытаний.

## Блочное и сквозное треугольное разложение матрицы

В табл. 1 показаны результаты решения СЛАУ (симметричная положительно-определенная матрица, размерностью 4,2 млн строк и 113 млн коэффициентов) стационарной задачи прочности (рис. 1) различными методами с кратким описанием отличий.

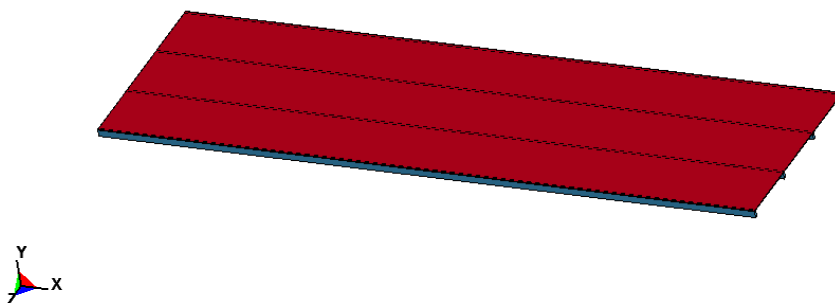


Рис. 1. Подкрепленная панель для моделирования кессонов самолета

Таблица 1

Решение СЛАУ P4\_6 из задачи прочности разными методами<sup>1</sup>

| Решатель               | p   | Dens Fct | N iter | Fct Time | Iter Time | Total Time |
|------------------------|-----|----------|--------|----------|-----------|------------|
| PARDISO Intel® MKL     | 12  | ~20      | 0      | –        | –         | 430        |
| MUMPS                  | 48  | ~20      | 20     | 115      | 2         | 117        |
| MUMPS+ <sup>2</sup>    | 48  | ~20      | –      | 90       | –         | 90         |
| PMLP_Schwartz(PARDISO) | 48  | <20      | 175    | 56       | 139       | 195        |
| PMLP_Schwartz(PARDISO) | 192 | <20      | 85     | 15       | 106       | 121        |
| VC_RAN_Schwartz(IC2)   | 49  | 4,0      | 4778   | 46       | 491       | 536        |
| VC_RAN_Schwartz(IC2)   | 385 | 4,3      | 6965   | 4        | 105       | 109        |
| TParBSS(IC2_float)     | 96  | 8,8      | 409    | 87       | 80        | 167        |
| TParBSS(IC2_float)     | 384 | 8,9      | 447    | 80       | 34        | 114        |

PARDISO Intel MKL традиционно решает задачи такого класса меньшего размера весьма эффективно. Но будучи ограниченным общей памятью, решает такую СЛАУ за 430,4 с, используя 12 ядер одного сервера и затрачивая ~ в 20 раз больше памяти, чем занимает сама матрица. MUMPS способен решить эту СЛАУ гораздо быстрее, используя несколько серверов, но будучи ограниченным 32-х разрядной адресно-индексной арифметикой не может решать СЛАУ размером в 4 раза больше. Проводя даже далеко не самое эффективное разбиение матрицы (и СЛАУ) на блоки, используя в PMLP/ParSol итерационный решатель CG, предобусловленный полным разложением (тем же PARDISO) в двусторонне перекрытых блоках, также решает эту СЛАУ гораздо быстрее на нескольких серверах и при этом размеры матрицы не ограничиваются. Такой же результат можно достичь, используя CG, предобусловленный односторонне перекрытым блочным неполным разложением IC2 (VC\_RAN\_SLAU), и сквозное неполное разложение IC2 (TParFBSS). При этом два последних способа гораздо меньше потребляют памяти (оба примерно одинаково, в TParFBSS – в 2 раза больше коэффициентов в треугольном факторе, но они 32-х разрядные и одновременно заблокированы по 9 штук).

Наряду с блочно-перекрытыми и сквозными треугольными разложениями допускается (в TParFBSS) сочетающее их предобусловливание, когда блоки MPI-процессов распределенной СЛАУ объединяются в группы блоков с расширением групп назад, и выполняется блочное по расширенным группам и сквозное в группах неполное разложение.

#### AMG Parallel Toolbox

Используется упрощенный алгоритм разделения узлов сетки на множество сильных и слабых узлов с использованием параметра силы связи, заданного регулируемым параметром  $\epsilon$ , интерполяция значений в слабом узле по значениям грубых узлов типа среднего арифметического. На основании этого строится оператор огрубления, грубые матрицы и реализуется V-цикл с операциями сглаживания Гаусса – Зейделя, содержащего релаксационный параметр, что проиллюстрировано на рис. 2 [10].

<sup>1</sup>Данные, представленные в таблице, предназначены для иллюстрации методов собранных в комплексе LParSol, являются характерными, но не самыми лучшими для каждой из библиотек.

<sup>2</sup>Адаптация для мелкоблочных матриц симметричного решателя MUMPS (ускорен ~ на 20 %) [11].

AMG Parallel Toolbox

упрощенный алгоритм Рунге-Штюбена [1,2] разделения ячеек на C и F, параметр  $\epsilon$  :

```

C ← ∅, F ← ∅, T ← I
while T ≠ ∅ do
  Find next node i ∈ T
  C ← C ∪ {i}
  F ← F ∪ {j ∈ I | j ∉ C ∪ F ∧ i ≠ j ∧ |Aij| > ε|Aii|}
  T ← T \ (C ∪ F)
end while
    
```

прямой сглаживатель Гаусса-Зейделя (+ *mcGS*, *ncycle*)

$u_i^p \leftarrow u_i^p + \omega \mathcal{D}^{-1} v_i^p$  - для граничных точек

$u_i^p \xleftarrow{\text{forward}} u_i^p + \mathcal{D}^{-1} (f_i^p - A_i^p u_i^p)$  - для внутренних точек

обратный сглаживатель Гаусса-Зейделя (+ *mcGS*, *nc*)

$u_i^p \xleftarrow{\text{backward}} u_i^p + \mathcal{D}^{-1} (f_i^p - A_i^p u_i^p)$  - для внутренних точек

$u_i^p \leftarrow u_i^p + \omega \mathcal{D}^{-1} v_i^p$  - для граничных точек

матрица интерполяции имеет вид [1,2]  
(+ адаптивно  $P=\text{const}$  в потоке):

$$P = \begin{pmatrix} \mathbf{1}_{CC} \\ P_{FC} \end{pmatrix}$$

где

$$(P_{FC})_{ij} := \begin{cases} 1/n_i, & |A_{ij}| > \epsilon|A_{ii}| \\ 0, & \text{else} \end{cases}$$

$$n_i := \#\{j \in C \mid |A_{ij}| > \epsilon|A_{ii}|\}$$

**AMG**: строим пирамиду:  $A_c = P^* \cdot A_r P; \dots;$

*щкд*:  $\{A_r x = r; \quad r_c = P \cdot r;$   
 $A_c \Delta x_c = r_c; \quad x_r = P \cdot \Delta x_c\}$

*пока*  $A_c < A_r$ .  
 (+ *изолир. f-точки не несём в C;*  
*огрубляем до конца последовательно*)

- Liebmann M. Algebraic Multigrid Methods on GPU-Accelerated Hybrid Architectures. – Institute for Mathematics and Scientific Computing, University of Graz, 2011.
- Курсивом отражены усовершенствования*

Рис. 2. Особенности AMG-предобусловливателя

В табл. 2 демонстрируется процесс построения AMG-предобусловливателя для одной из СЛАУ из двумерной задачи теплопроводности. Как видно, матрица огрубляется 10 раз (пирамида из 11 матриц), 8 из которых выполнялись в *последовательном режиме*<sup>3</sup> (определяется параметром перехода). Здесь  $nV$  – число вершин графа (строк матрицы),  $nE$  – число ребер (коэффициентов матрицы). Отметим, что а) исходная распределенная СЛАУ *односторонне однослойно перекрывается и уравнения в перекрытии (коэффициенты матрицы и правой части) особым образом расщепляются*; б) строка «0 (по MPI)» означает состояние распределенной перекрытой огрубленной матрицы перед сборкой ее на одном MPI – процессе, которая завершается исключением перекрытий (доля которых в общем объеме матрицы – увеличивается с повышением уровня).

Таблица 2

Результаты редукции матрицы ( $nV$  – число и  $nE$  – число ненулевых элементов)

| $nLevel$ ( $nMPI$ ) | $nV$                             | $nE$                              | $nE/nV$                | $nV_i/nV_{i-1}$          |
|---------------------|----------------------------------|-----------------------------------|------------------------|--------------------------|
| 0 (6 MPI)           | 1271479                          | 6230247                           | 4,9                    |                          |
| 1 (6 MPI)           | 604075                           | 5376276                           | 8,9                    | 0,48                     |
| 2 (6 MPI)           | 146487                           | 1414639                           | 9,6                    | 0,24                     |
| 3 (1 MPI)           | 32091                            | 351351                            | 10,9                   | 0,22                     |
| 4, 5, 6,7 (1 MPI)   | 6627, 1594,<br>482, 188          | 76837, 15792,<br>3108, 928        | 11,6, 9,9,<br>6,4, 4,9 | 0,21, 0,24,<br>0,3, 0,39 |
| 8 (1 MPI)           | 88                               | 338                               | 3,8                    | 0,47                     |
| 9 (1 MPI)           | 38                               | 88                                | 2,3                    | 0,43                     |
| 10 (1 MPI)          | 7                                | 7                                 | 1                      | 0,18                     |
| $\Sigma$            | $(nV_\Sigma/nV_0 = 1,6)$ 2063156 | $(nE_\Sigma/nE_0 = 2,2)$ 13469611 | 6,5                    |                          |

<sup>3</sup>Курсивом выделены усовершенствования.

На каждом уровне выполнялось одно пред- и пост-сглаживание<sup>4</sup> (в сумме по всем уровням ~54 млн операций на  $1V$ -цикл). Для сокращения начальной невязки в 1 млн раз AMG-решателю потребовалось 29 итераций ( $V$ -циклов, 1600 млн операций). Решателю *BiCGStab* с AMG-предобусловливателем потребовалось 7 итераций (~550 млн операций)<sup>5</sup>, что гораздо надежнее и экономичней.

В табл. 3 приведены результаты проверки масштабируемости решателя *BiCGStab* предобусловленного методом AMG той же СЛАУ в режиме линейной (не самой выгодной) декомпозиции. В табл. 4 отражено большое ускорение счета стационарной задачи двухфазной фильтрации в программном комплексе НИМФА [17] по сравнению с другим применяющимся предобусловливателем.

Таблица 3

Таймирование AMG(...) $\cdot$ BiCGStab; точность  $10^{-8}$ , режим week scaling

|                               |      |      |      |      |      |      |
|-------------------------------|------|------|------|------|------|------|
| $n_{MPI} \cdot 1$ ядро        | 1    | 2    | 8    | 64   | 128  | 256  |
| $nV/n_{MPI}$ , тысяч          | 1271 | 636  | 159  | 20   | 10   | 5    |
| $n_{Iter}$                    | 11   | 11   | 11   | 11   | 10   | 11   |
| $t_{Setup}$ , с               | 1,71 | 1,07 | 0,31 | 0,10 | 0,15 | 0,26 |
| $t_{Setup}$ , $P = const$ , с | 0,54 | 0,29 | 0,11 | 0,01 | 0,01 | 0    |
| $t_{Iter}$ , с                | 4,00 | 2,65 | 1,30 | 0,17 | 0,10 | 0,08 |
| $t_{All}$ (соседи даны), с    | 5,71 | 3,72 | 1,61 | 0,27 | 0,25 | 0,34 |
| $t_{All}$ , $P = const$ , с   | 4,54 | 2,94 | 1,41 | 0,18 | 0,10 | 0,08 |
| Eff, %                        |      | 77 % | 44 % | 33 % | 18 % | –    |
| Eff, $P = const$ , %          |      | 77 % | 40 % | 49 % | 35 % | 22 % |

Таблица 4

Таймирование решателей в НИМФА; 36 MPI-процессов в задаче

| Решатель СЛАУ                     | Размер сетки | $t_{решения}$ , с<br>(% от всей задачи) | $\Sigma_{шагов}/\Sigma_{СЛАУ}$ | Число итераций.<br>Общее/среднее/max |
|-----------------------------------|--------------|---|--------------------------------|--------------------------------------|
| AMG $\cdot$ BiCGStab              | 285180       | 3808,14 (29,05)                         | 1730/5135                      | 15258/3/15                           |
| BJac(ILU0_float) $\cdot$ BiCGStab | 285180       | 13995,09 (64,73)                        | 1720/3448                      | 304024/88/441                        |

При увеличении шага задачи по времени число итераций решателя с AMG (в противоположность с ILU0) слабо возрастало. Текущая версия AMG не безразлична к классу задач и во многих случаях затратнее ILU-базированного предобусловливателя, но, где работает, там экономичнее ILU0 (по времени).

### Коррекция решения на грубой сетке

Цели коррекции (уточнения) решения на грубой сетке (Coarse Grid Correction – CGC) – сокращение низкочастотной составляющей ошибки и ее рассогласованности в подобластях. CGC,

<sup>4</sup>Одно сглаживание Гаусса – Зейделя  $\sim 2 \cdot E_l$  арифметических операций двойной точности.

<sup>5</sup>На одной итерации BiCGStab выполняется  $\sim 2 \cdot 2 \cdot nE_0$  операций и  $2V$ -цикла.

как известно, состоит из двух этапов: огрубление исходной матрицы и решение огрубленной СЛАУ. На первом этапе вычисляются: матрица пролонгации грубого решения  $\mathbf{P}$  размером  $N \times n$  и грубосеточная матрица  $\mathbf{A}_c = \mathbf{P}^T \mathbf{A} \mathbf{P}$ , где  $\mathbf{A}_c$  – грубая матрица  $n \times n$ ,  $\mathbf{A}$  – исходная матрица  $N \times N$ .

В нашей реализации  $\mathbf{P}$  строится рекурсивно отождествлением (обычно двух) связанных неизвестных и слиянием их в одно грубое неизвестное ( $i$  присоединяет  $j$ , если  $|A_{ij}| \geq |A_{ik}|$  для любого  $k \neq i$ ).<sup>6</sup> Рекурсия останавливается, когда число грубых неизвестных в каждой подобласти (в каждом MPI-процессе) сократится до заданного пользователем числа (по умолчанию 1 + число соседних MPI-процессов). Получившаяся грубая матрица  $\mathbf{A}_c$  используется в предобусловливателе. Построенный предобусловливатель  $\mathbf{M}$  с CGC-оператором уточнения  $\mathbf{P} \mathbf{A}_c^{-1} \mathbf{R}$ , которому на вход подан вектор  $\mathbf{r}$ , применяется следующим образом:

$$\Delta \mathbf{x} = (\mathbf{P} \mathbf{A}_c^{-1} \mathbf{R}) \mathbf{r} - \text{нашли грубую поправку (сократили низкочастотную ошибку),}$$

$$\Delta \mathbf{r} = \mathbf{r} - \mathbf{A} \Delta \mathbf{x} - \text{вычислили новую невязку,}$$

$$\Delta \mathbf{x} = \Delta \mathbf{x} + \mathbf{M}^{-1} \Delta \mathbf{r} - \text{сгладили (сократили) высокочастотную ошибку.}$$

Подзадача  $\mathbf{y} = \mathbf{A}_c^{-1} \mathbf{f}$ , решается в виде  $\mathbf{A}_c \mathbf{y} = \mathbf{f}$ , используя параллельный прямой решатель или итерационный решатель, предобусловленный  $\text{ILUt}(\mathbf{A}_c)$ . CGC может использоваться в PMLP/ParSol для решения СЛАУ с одной неизвестной в ячейке сетки в качестве предобусловливателя ( $\mathbf{M} = \mathbf{I}$ ) и в сочетании с предобусловливателем Шварца и сквозного  $\text{ILUt}$ .

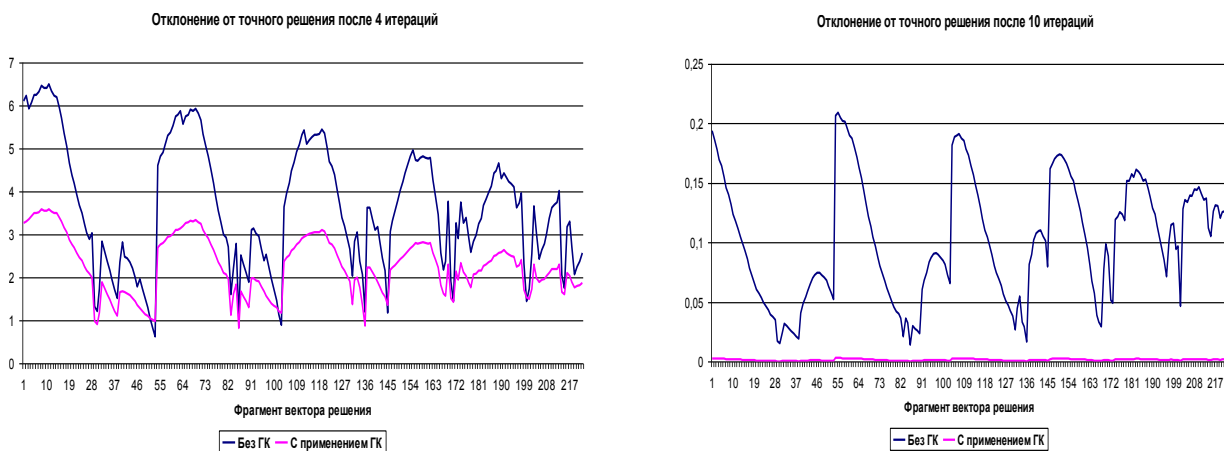


Рис. 3. Отклонение от точного решения CGC•Schwartz(...) $\cdot$ BiCGStab

Эксперименты с CGC•Schwartz(ILU...) на разных СЛАУ из программ расчета теплопроводности ВЦ ВНИИЭФ показали: а) сходимость, как правило, не хуже, чем с Schwartz(ILU...), б) сходимость повышается там же, где применение AMG находит хороший отклик, в) CGC•Schwartz(ILU) сокращает время решения СЛАУ, когда без CGC число итераций – десятки.

Например, комплекс МЕДУЗА-П, решая задачу за 15 мин посчитал:

– 139 шагов с решателем  $\text{Schwartz}(\text{ILUt}(0.001), 4 \text{ слоя}, \dots) \cdot \text{BiCGStab}$ ;

– <139 шагов, используя  $\text{CGC}(\dots) \cdot \text{Schwartz}(\dots) \cdot \text{BiCGStab}$  для всех СЛАУ;

– >200 шагов, используя  $\text{CGC}(\dots) \cdot \text{Schwartz}(\dots) \cdot \text{BiCGStab}$  для решения только первых двух СЛАУ на каждом временном шаге (остальные СЛАУ – без CGC).<sup>7</sup>

<sup>6</sup>Подобно агрегативному AMG, хорошо себя зарекомендовавшему в гидродинамике [4].

<sup>7</sup>При этом число итераций  $\text{BiCGStab}$  на первой СЛАУ снизилось со 170 до 37 штук (время решения с 2,1 до 1,4 с), на второй – со 103 до 23 (время решения 1,2 до 0,8 с).

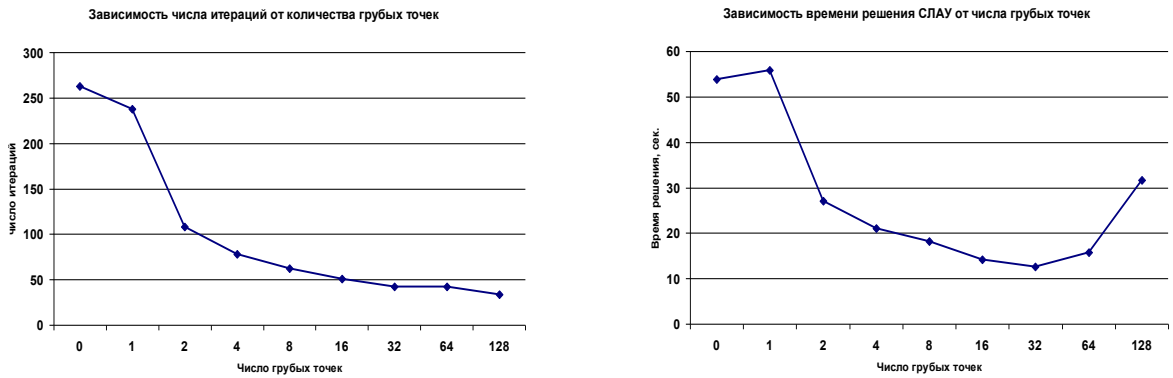


Рис. 4. Число итераций и время решения СЛАУ tube\_ground из модуля SIMPLE комплекса ЛОГОС

### Точки роста

В настоящее время становятся все более востребованными MPI•OpenMP решатели. Техника счета на множестве нитей освоена в библиотеке TParFBSS на основе Intel TBB и PMLP/ParSol на основе pthreads. Здесь мы столкнулись с трудностями включения таких решателей в приложения из-за возросших требований к комплектации системного ПО и начали разработку собственных мультипоточных решателей на основе OpenMP – стандартного средства распараллеливания на многоядерных вычислительных устройствах с общей памятью.

В PMLP/ParSol разрабатываются следующие мультипоточные возможности:

- ✓ 32/64-разрядное предобусловливание 64/64 + разрядных решателей;
- ✓ раскраска, переупорядочивание графа;
- ✓ построение конвейера независимых вычислений в треугольных системах;
- сужение полуширины, огрубление, декомпозиция графа;
- транспонирование матрицы;
- умножение разреженных матриц оператора  $P^tAP$ ;
- векторизация вычислений с блоками чисел одинакового размера;

и мультипоточные алгоритмы (решатели и предобусловливатели):

- ✓ CG, BiCGStab, ...,
- ✓ mcGS, mcSGS, mcILU0, pipeILU0, pipeILDU0,
- ✓ blockILUt/ICt/ICk/ILUk,
- block\_overlapILUt/ICt/ICk/ILUk,
- CGC,
- AMG.

Уже реализованы (табл. 5) внутриобластные мультипоточные итерационные решатели CG, BiCGStab и легкие предобусловливатели – mcILU0, pipeILU0, blockILU.../IC... – соответственно многоцветный (multi color), конвейерный (level scheduling [18]) и блочно-неперекрытый ILU, охватывающие все возрастающую многоядерность вычислительных устройств.

Рассматриваются варианты реализации дополнения Шура, поскольку на практике встречаются СЛАУ, требующие сквозного ILUt для сдерживания роста итераций с увеличением разбиения СЛАУ. Один из способов [19] базируется на точном разложении корневого дополнения Шура. Здесь, так же как и в предобусловливателе Шварца, было бы полезным использование разрабатываемого в ННГУ мультипоточного прямого решателя, мультипоточный  $LL^t$ -вариант которого конкурирует по производительности с MUMPS.

Таблица 5

Сравнение MPI•OMP решателей JacobiBlock(метод)•BiCGStab; 8 MPI-процессов на 8 узлах по 16 ядер; 1 млн уравнений, 26 млн коэффициентов

| Метод            | $n$ Threads | All time | Setup time | Iter time | $n$ Iter |
|------------------|-------------|----------|------------|-----------|----------|
| blockILUt(0,001) | 1           | 2,09     | 1,08       | 1,00      | 18       |
|                  | 16          | 0,36     | 0,12       | 0,23      | 23       |
| blockILU0_float  | 1           | 0,74     | 0,24       | 0,49      | 18       |
|                  | 16          | 0,22     | 0,07       | 0,15      | 22       |
| pipeILU0         | 1           | 1,67     | 0,33       | 1,34      | 18       |
|                  | 16          | 0,39     | 0,07       | 0,32      | 18       |
| mcILU0           | 1           | 1,58     | 0,66       | 0,92      | 20       |
|                  | 16          | 0,28     | 0,05       | 0,23      | 20       |

### Литература

1. Dongarra J., Hittinger J., Bell J. et al. Applied Mathematics Research for Exascale Computing (Exascale Mathematics Working Group) / LLNL TR-651000, 2014.
2. Капорин И. Е. Дискретный метод наименьших квадратов для полиномиального преобусловливания несимметричных матриц / Под ред. В. Г. Жадана // Оптимизация и приложения: Сб. трудов отдела проблем прикладной оптимизации ВЦ РАН. Вып. 3. М.: Изд-во ВЦ РАН, 2013. С. 120–139.
3. Коньшин И. Н., Сушко Г. Б., Харченко С. А. и др. Решение плохообусловленных разреженных СЛАУ большого размера с помощью PARFBSS / Под ред. Р. М. Шагалиева // Труды XIII Международ. конф. «Супервычисления и математическое моделирование». Саров: РФЯЦ-ВНИИЭФ, 2012. С.76–78.
4. Козелков А. С., Дерюгин Ю. Н., Лашкин С. В. и др. Реализация метода расчета вязкой несжимаемой жидкости с использованием многосеточного метода на основе алгоритма SIMPLE в пакете программ ЛОГОС // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2013. Вып. 4. С. 44–56.
5. Жуков В. Т., Новикова Н. Д., Феодоритова О. Б. Ультрапараллельный многосеточный метод для эллиптических уравнений / Под ред. Р. М. Шагалиева // Труды XIII Международ. конф. «Супервычисления и математическое моделирование». Саров: РФЯЦ-ВНИИЭФ, 2012. С. 92–93.
6. Бартенев Ю. Г., Бондаренко Ю. А., Ерзунов В. А. и др. Комплекс LParSol для решения СЛАУ / Под ред. Р. М. Шагалиева // Там же. С. 34–36.
7. Intel Math Kernel Library Reference Manual [Electronic resource]. URL. Mode of access: <http://software.intel.com/sites/products/documentation/hpc/mkl/mklman.pdf>.
8. Henson V. E., Yang U. M. BoomerAMG: a Parallel Algebraic Multigrid Solver and Preconditioner // Applied Numerical Mathematics. 2002. 41. P. 155–177.
9. MUltifrontal Massively Parallel Solver (MUMPS 4.10.0) User's guide // Technical report ENSEEINT-IRIT. 2011. [Electronic resource]. URL. Mode of access: [http://mumps.enseeiht.fr/doc/user-guide\\_4.10.0.pdf](http://mumps.enseeiht.fr/doc/user-guide_4.10.0.pdf).
10. Liebmann M. Algebraic Multigrid Methods on GPU-Accelerated Hybrid Architectures // Institute for Mathematics and Scientific Computing, University of Graz, 2011.
11. Лебедев С. А., Мееров И. Б., Сысоев А. В. и др. Оптимизация и применение пакета MUMPS для решения трехмерных стационарных задач прочности на кластерных системах // Вестник УГАТУ. 2014. Принято к печати.
12. Бартенев Ю. Г., Карпов А. П., Крючков И. А., Пишулин И. А. Адаптация некоторых решателей СЛАУ для арифметических ускорителей NVIDIA / Под ред. Р. М. Шагалиева // Труды XIV Международ. конф. «Супервычисления и математическое моделирование». Саров: РФЯЦ-ВНИИЭФ, 2013. С. 30–32.



13. Коньшин И. Н., Сушко Г. Б., Харченко С. А. Сквозной параллельный алгоритм построения неполного треугольного разложения второго порядка точности с динамическим выбором декомпозиции и упорядочивания // Тез. докл. XIV Международ. конф. «Супервычисления и математическое моделирование». Саров, 2012. / Под ред. Р.М. Шагалиева. Труды XIV Международ. конф. «Супервычисления и математическое моделирование». Саров: РФЯЦ-ВНИИЭФ, 2013. С. 110–111.

14. Ерзунов В. А., Горбунов А. А. Механизм адаптивного выбора решателя в библиотеке PMLP/Parsol // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2009. Вып. 1. С. 55–62.

15. Капорин И. Е., Милукова О. Ю. Предобусловливание итерационных методов для эффективного массивно-параллельного решения систем линейных алгебраических уравнений / Под ред. Р. М. Шагалиева // Труды XIII Международ. конф. «Супервычисления и математическое моделирование». Саров: РФЯЦ-ВНИИЭФ, 2012. С. 71–72.

16. Smith B. F., Bjorstad P. E., Gropp W. D. Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge: Cambridge University Press, 2004.

17. Бутнев О. И., Пронин В. А., Сидоров М. Л. и др. Пакет программ НИМФА-2 для решения задач многофазной фильтрации с применением суперкомпьютерных технологий / Под ред. Р. М. Шагалиева // Труды XIV Межд. конф. «Супервычисления и математическое моделирование». Саров, 2013. С. 112–119.

18. Saad Y. Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

19. Lukarski D., Anzt H., Tomov S., Dongarra J. *Multi-Elimination* ILU Preconditioners on GPUs: Technical Report UT-CS-14-723 / Innovative Computing Laboratory, University of Tennessee, 2014.

## **ПЕРСПЕКТИВНАЯ ГИБРИДНАЯ ТОПОЛОГИЯ KNS ДЛЯ КОММУНИКАЦИОННОЙ СЕТИ НА ОСНОВЕ АППАРАТНОГО МОДУЛЯ СМПО-10GA-1**

*В. Г. Басалов*

Российский федеральный ядерный центр –  
Всероссийский НИИ экспериментальной физики, г. Саров

### **Введение**

Около двадцати лет в ВНИИЭФ создаются суперкомпьютеры семейства МП-Х-У, как с использованием коммерческих коммуникационных сетей (Mugicom, InfiniBand), так и с использованием систем межпроцессорного обмена собственной разработки в суперкомпьютерах МП-3 [1], МП-3Т, МП-СМПО-2D и МП-СМПО-3D.

В настоящее время в ИТМФ разработана отечественная система межпроцессорного обмена СМПО-10GA-1. Областью применения аппаратных компонент СМПО-10GA-1 является создание высокопроизводительных коммуникационных сетей для вычислительных систем разного уровня производительности: от компактных суперЭВМ до больших систем, состоящих из тысяч вычислительных узлов. 64-узловая вычислительная система МП-СМПО-3D, созданная на базе СМПО-10G-1 с топологией MultiTor, выдержала тестовые испытания.