

Масштабируемость гибридного MPI+OpenMP AMG решателя

(Distributed and hybrid-memory AMG with extended prolongation stencil)

И. Н. Коньшин, К. М. Терехов

ИВМ РАН



28-29 сентября 2026 г.

MPI

Масштабируемость распределённого AMG

(Reaching the scalability of a distributed AMG solver)

- Системы линейных уравнений достигают миллиардов неизвестных (нефтяные резервуары, геомеханика).
- Требуются распределённые суперкомпьютеры с тысячами узлов.
- Наиболее эффективный метод для эллиптических задач — **алгебраический многосеточный метод (AMG)**.
- Однако эффективная реализация AMG на распределённой памяти остаётся сложной задачей.

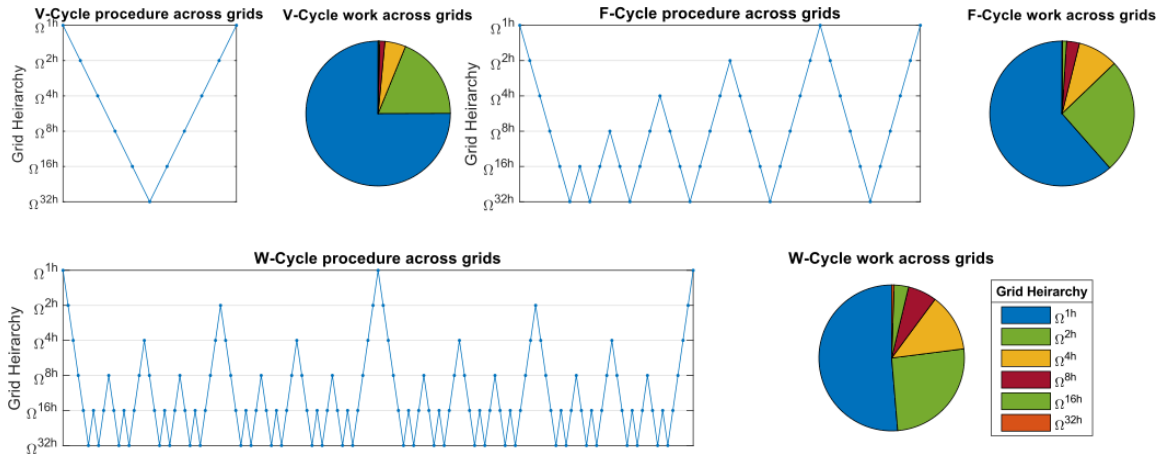
Основные компоненты AMG

- Иерархия подпространств: операторы пролонгации P_k и сужения $R_k = P_k^T$
- Матрица на следующем уровне: $B_k = R_k A_k P_k$
- Сглаживатели: Чебышев, Гаусс–Зейдель, ℓ_1 -Jacobi
- Прямое решение на самом грубом уровне: LU-факторизация
- Последовательность решения систем: V-, W-, F-, L-циклы
- Сокращение коммуникатора на грубых уровнях

Структурные свойства тестовых матриц

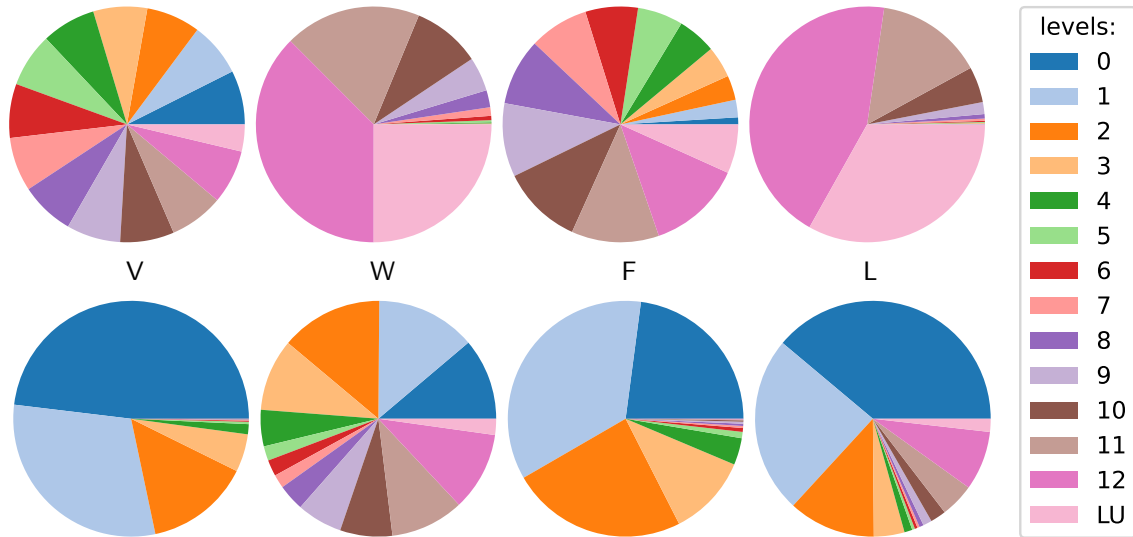
Матрица	N	N_{nz}	N_{nz}/N	Происхождение
Cube5317k	5 317 443	222 615 369	41.9	однородная упругость
Queen_4147	4 147 110	329 499 288	79.5	структурная механика
Stocf_1465	1 465 137	21 005 389	14.3	пористая среда
Guenda11m	11 452 398	512 484 300	44.7	геомеханика
Agg14m	14 106 408	633 142 730	44.9	мезомасштабное моделирование
M20	20 056 050	1 634 926 088	81.5	механика
Rtanis44m	44 798 517	747 633 815	16.7	пористая среда
Utemp20m	20 113 744	242 765 052	12.1	гетерогенная температурная диффузия
Pflow73m	73 623 733	2 201 828 891	29.9	пористая среда
L2d	225 000 000	2 024 820 004	9.0	2D ур-е Пуассона (9-точечный шаблон)
L3d	100 000 000	2 678 425 192	26.8	3D ур-е Пуассона (27-точечный шаблон)
P512	134 217 728	937 951 232	7.0	3D ур-е Пуассона (7-точечный шаблон)
P1024	1 073 741 824	7 509 901 312	7.0	3D ур-е Пуассона (7-точечный шаблон)
P1536	3 623 878 656	25 352 994 816	7.0	3D ур-е Пуассона (7-точечный шаблон)
Cx64(2048)	536 870 912	3 753 902 080	7.0	3D ур-е Пуассона (7-точечный шаблон)
Cx256(128)	2 147 483 648	15 021 899 776	7.0	3D ур-е Пуассона (7-точечный шаблон)

Сравнение циклов V,W,F (wiki)



- V-цикл — простой, но число итераций растёт с размером задачи.
- W- и F-циклы независимы от размера, но менее масштабируемы.

N_{it} и T для циклов V,W,F,L на P1024 (64 nodes = 896 cores)



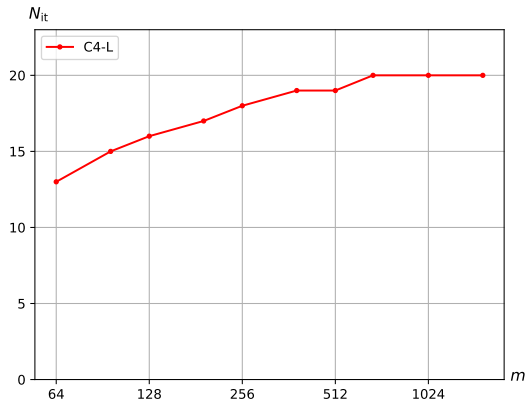
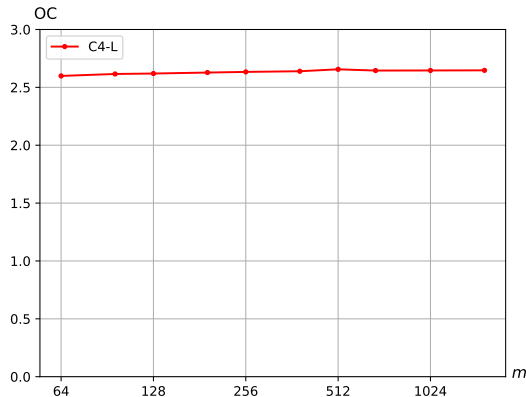
Количество итераций N_{it} от размера задачи для CH циклов

	CH-V	CH-W	CH-F	CH-L
P256	21	7	19	19
P512	27	7	13	20
P1024	34	7	15	21

- V-цикл – рост довольно сильный :(
- W-цикл – постоянное число итераций !!
- F-цикл – по разному, но скорее будет меняться мало...
- L-цикл – меняется мало

Предложенный L-цикл на задаче P512 (896 ядер)

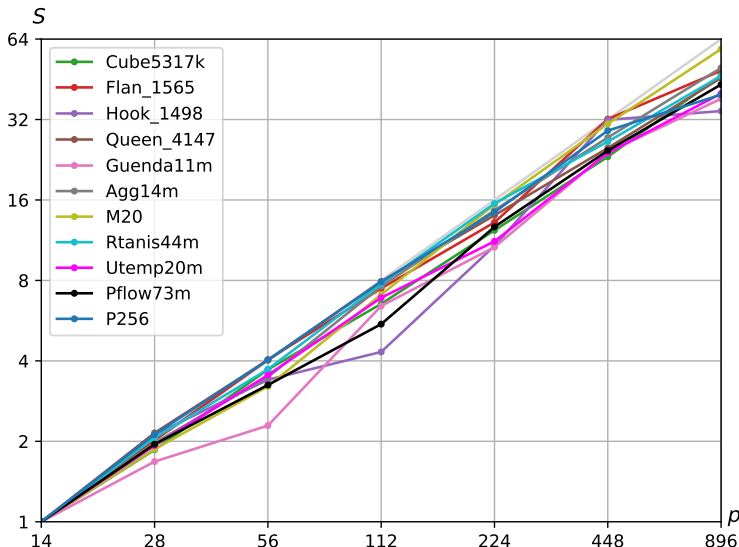
- Наименьшее полное время T (среди циклов V,W,F,L).
- Операторная сложность $OC \approx 2.66$ постоянна.
- Количество итераций N_{it} слабо зависит от размерности: При росте размерности от 64^3 до 1536^3 количество итераций увеличивается менее чем на 1 при увеличении размера в 2 раза.



Сравнение с PETSc GAMG и HYPRE BoomerAMG (P512)

Solver	N_{it}	L	OC	T_s	T_{it}	T	Remark
PETSc	9	5	1.58	1.89	0.75	2.64	
HYPRE	15	10	2.77	2.12	1.99	4.11	
GS-V	27	11	2.66	1.04	2.16	3.19	*
GS-W	7	11	2.66	1.09	16.02	17.11	
GS-F	12	11	2.66	1.08	3.17	4.25	
GS-L	20	11	2.66	1.07	2.88	3.95	*
CH-V	27	11	2.66	0.63	2.13	2.76	*
CH-W	7	11	2.66	0.64	5.38	6.02	
CH-F	13	11	2.66	0.65	2.37	3.03	*
CH-L	20	11	2.66	0.63	1.96	2.59	**
C4-V	26	11	2.66	0.63	2.10	2.73	*
C4-W	6	11	2.66	0.64	4.71	5.35	
C4-F	12	11	2.66	0.65	2.24	2.89	*
C4-L	19	11	2.66	0.63	1.80	2.43	***

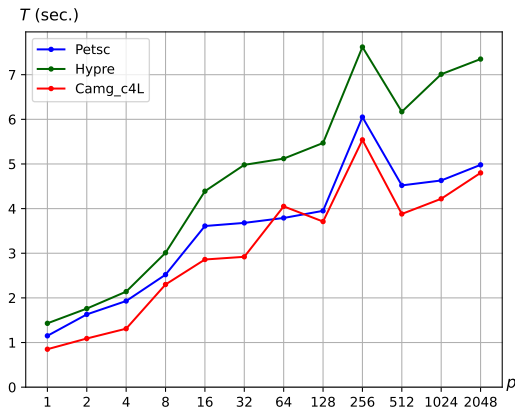
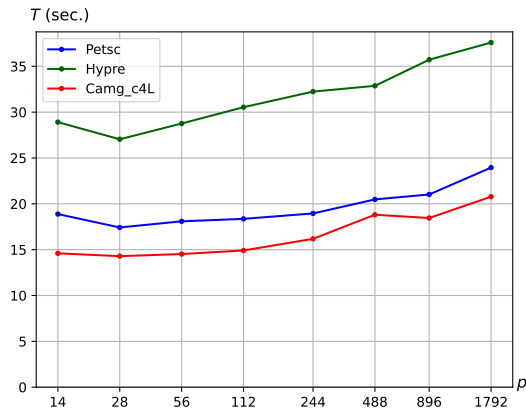
Сильная масштабируемость (Speedup) для C4-L



- Для задач от 1 до 70 млн неизвестных на 64 узлах (896 ядер) эффективность $> 50\%$, для половины задач $> 70\%$.
- Предложенный решатель решил задачу M20 на одном узле, тогда как PETSc/HYPRE не смогли из-за нехватки памяти.

Слабая масштабируемость: $Sx256(P)$ и $Sx64(p)$

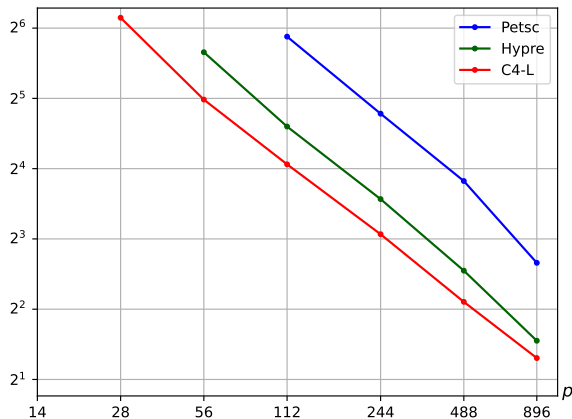
- $Sx256(P)$ на каждом узле по 256^3 неизвестных. При $P = 128$ в сумме > 2 млрд неизвестных.
- $Sx64(p)$ на каждом узле по 64^3 неизвестных. При $p = 2048$ в сумме > 500 млн неизвестных.
- Для C4-L на $Sx256(P)$ эффективность $E_1 \approx 70\%$ на 1792 ядрах.



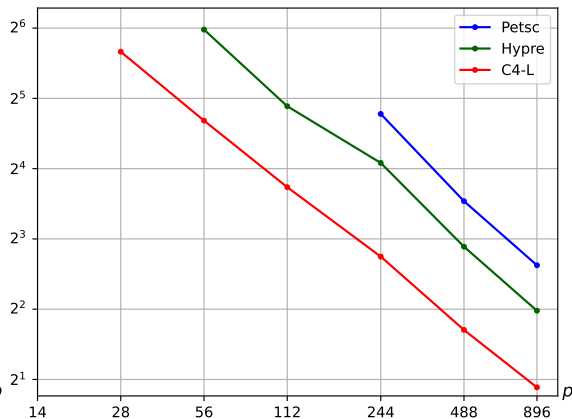
Сильная масштабируемость: L2d и L3d

- L2d (225 млн), L3d (100 млн): C4-L быстрее PETSc и HYPRE в 2–3 раза.
- C4-L решает с 2 узлов, PETSc — с 4, HYPRE — с 4–16 узлов из-за памяти.

T (sec.)

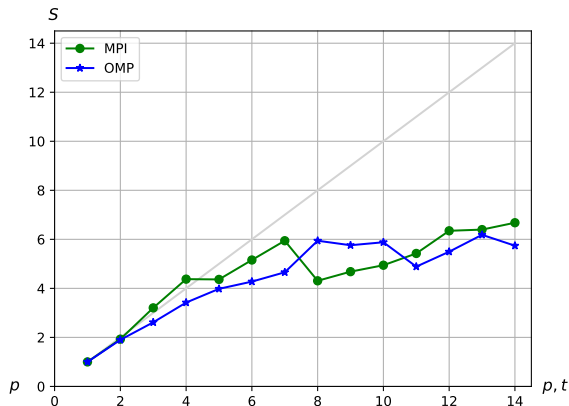
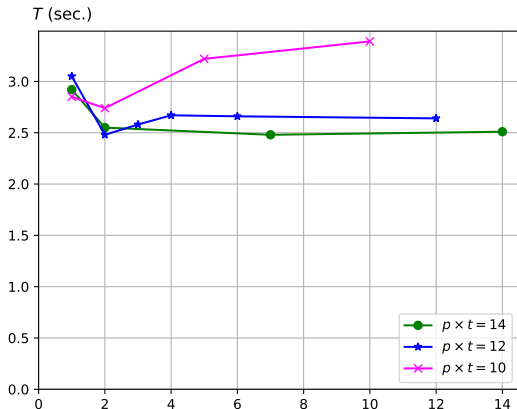


T (sec.)



Гибридная MPI+OpenMP версия AMG

- Исследована на P512 (64 узла, до 14 ядер на узел).
- MPI даёт большее ускорение, чем OpenMP, из-за лучшей локальности данных.
- Гипертрединг не улучшает эффективность.



Выводы по распределенной версии AMG

- Предложен эффективный параллельный AMG с L-циклом, сглаживателем C4 и динамическим сокращением коммуникатора.
- Метод C4-L превосходит PETSc и HYPRE по:
 - времени решения (T)
 - памяти (OC)
 - масштабируемости (S)
- Исследования на задачах до размерности 3.6 млрд. и кол-ва ядер до 2048.

OpenMP

Масштабируемость AMG на общей памяти

(Improving the robustness and scalability of shared-memory AMG solvers for point-block problems)

Структурные свойства тестовых матриц

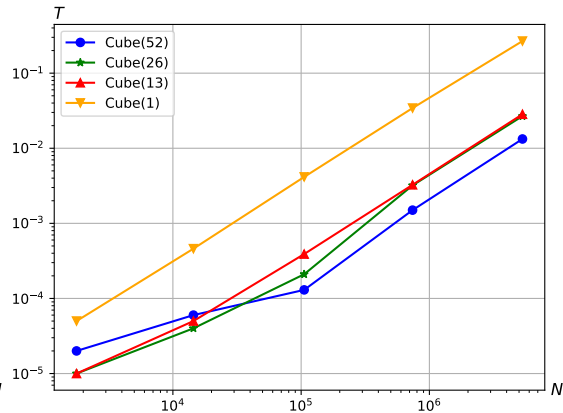
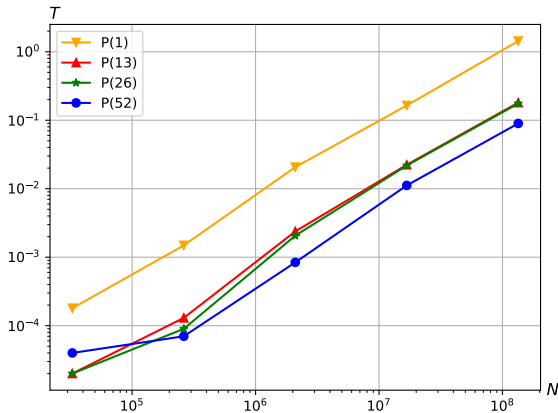
Матрица	N	N_{nz}	N_{nz}/N	Происхождение
Cube1k	1 773	63 927	36.1	однородная упругость
Cube14k ...	14 460	560 970	38.8	однородная упругость
Cube739k	739 167	29 657 925	40.1	однородная упругость
Cube5317k	5 317 443	222 615 369	41.9	однородная упругость
Agg14m	14 106 408	633 142 730	44.9	мезомасштабное моделирование
M20	20 056 050	1 634 926 088	81.5	механика
Tripod24m	24 186 993	1 111 751 217	46.0	механика
P32	32 768	223 232	6.8	3D ур-е Пуассона (7-точечный шаблон)
P64 ...	262 144	1 810 432	6.9	3D ур-е Пуассона (7-точечный шаблон)
P256	16 777 216	117 047 296	7.0	3D ур-е Пуассона (7-точечный шаблон)
P512	134 217 728	937 951 232	7.0	3D ур-е Пуассона (7-точечный шаблон)
Ddif1 ...	819	2 773	3.4	2D адаптивное квадродерево
Ddif8	979 797	3 029 311	3.1	2D адаптивное квадродерево
Knob1 ...	1 118	2 772	2.5	2D треугольная адаптивная сетка
Knob8	1 148 318	2 878 300	2.5	2D треугольная адаптивная сетка

Выбор лучшего параллельного алгоритма для AMG

- Выбор максимально независимого подмножества: $\mathcal{M} = \{M_{\text{mis}}, M_{\text{pmwis}}, M_{\text{pmis}}\}$
- Выбор оператора продолжения: $\mathcal{C} = \{C_{\text{rs}}, C_{\text{min}}, C_{\text{all}}, C_{\text{sum}}, C_{\text{infl}}\}$
- Применение агрессивного грубления: $\mathcal{A} = \{A_{\text{sqg}}, A_{\text{mis2}}\}$
- Метод сглаживания: $\mathcal{S} = \{S_{\text{hgs}}, S_{\text{ch4}}\}$
- Выбор индикатора для блоков матрицы: $\mathcal{I} = \{I_{\text{fb}}, I_{\text{tr}}, I_{\text{svd}}\}$
- Алгоритмы для операций с разреженными матрицами: $A \cdot x$, $R \cdot A \cdot P$, A^T
- Выбраны лучшие алгоритмы для $t = 1, 13, 26, 52$ нитей OpenMP

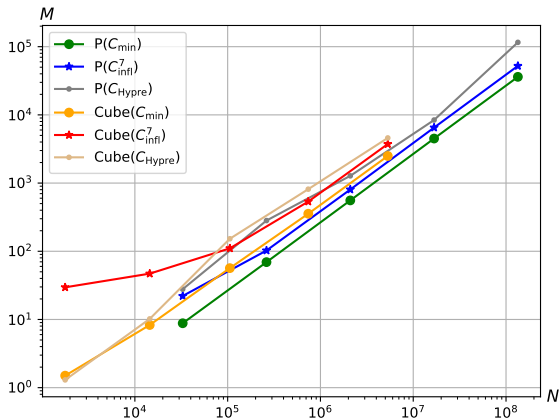
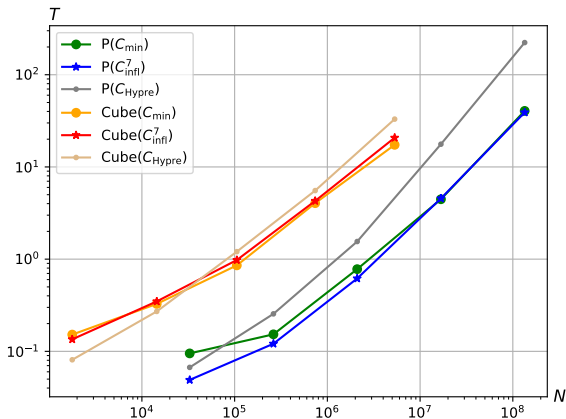
Время умножения разреженной матрицы A на вектор

- Серии Poisson и Cube для $t = 1, 13, 26, 52$ нитей OpenMP.



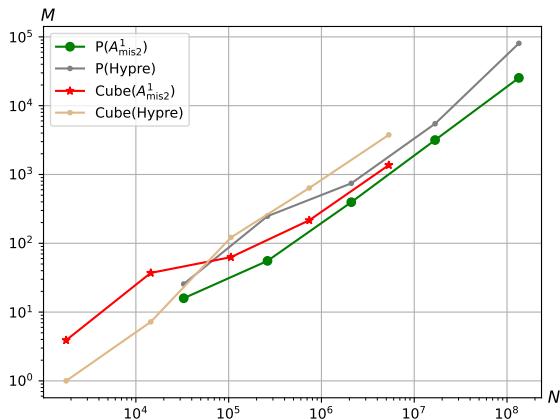
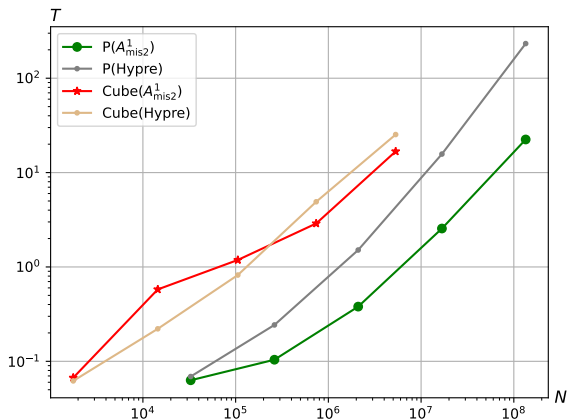
Время и память для различных стратегий $C_{\min}$, C_{infl}^7 , Hypre (оператор продолжения)

- Серии Poisson и Cube для $t = 1, 13, 26, 52$ нитей OpenMP.



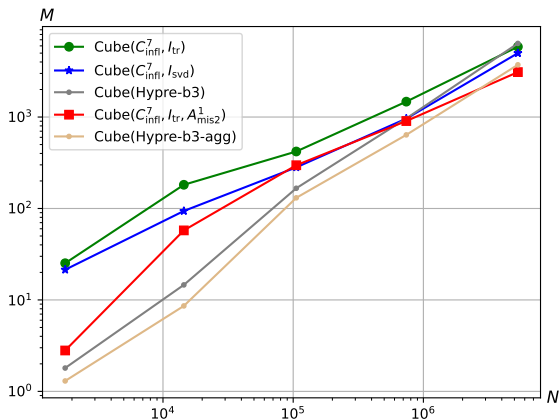
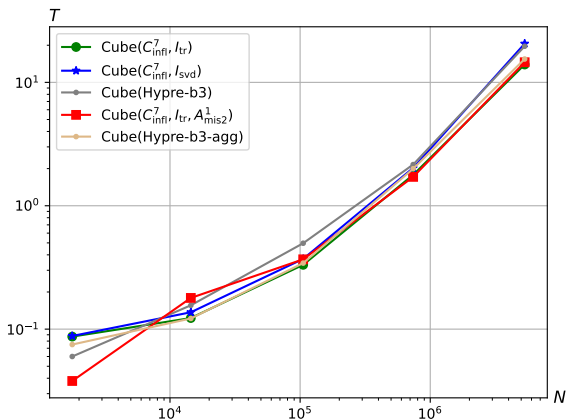
Время и память для стратегий C_{infl}^7 с A_{mis2}^1 и Hypre (продолжение с грублением)

- Серии Poisson и Cube для $t = 1, 13, 26, 52$ нитей OpenMP.

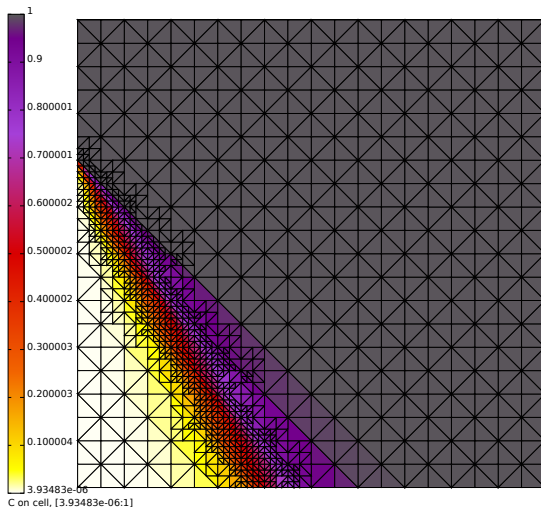
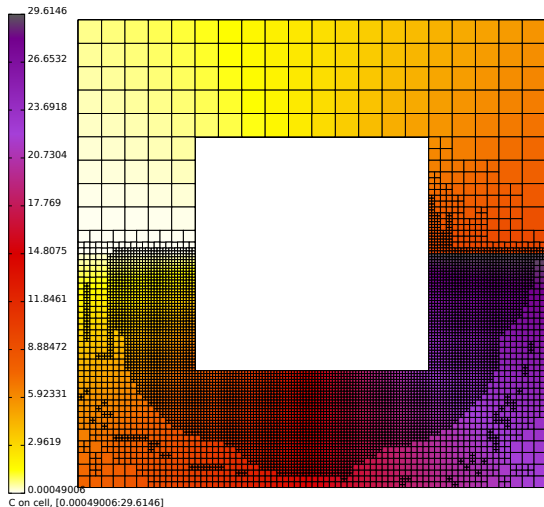


Время и память для C_{infl}^7 с I_{tr} , I_{svd} , $I_{\text{tr}} + A_{\text{mis2}}^1$ и Hypre (продолжение для блочных матриц)

- Серия Cube для $t = 1, 13, 26, 52$ нитей OpenMP ($b = 3$).

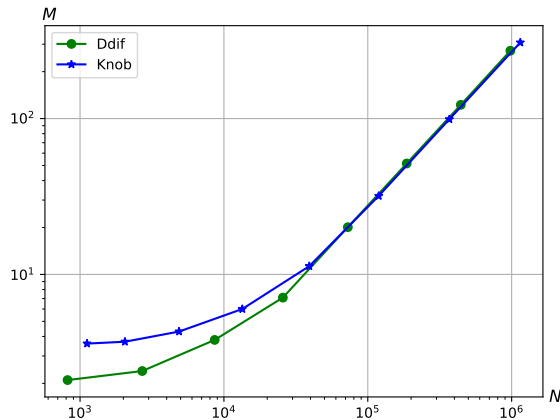
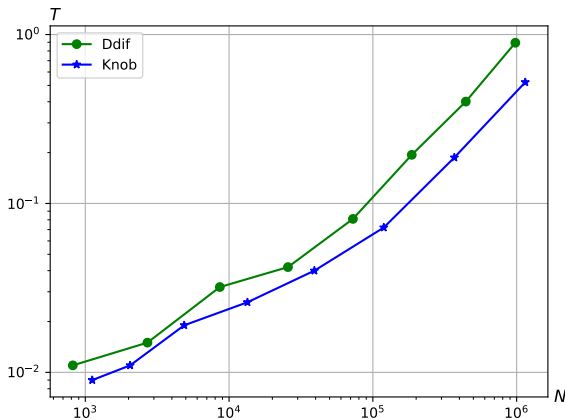


Адаптивные сетки для задач Ddif3 и Knob2



Время и память для серий Ddif и Knob на OpenMP($t=26$)

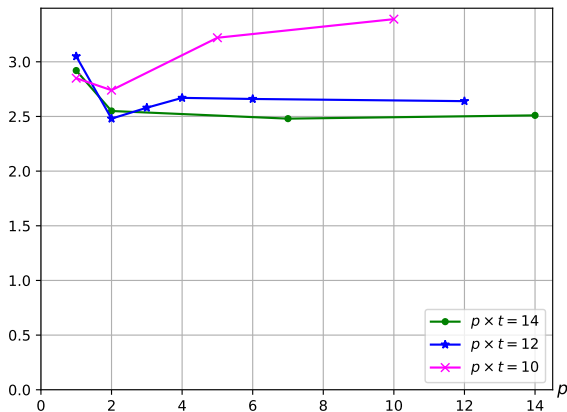
- Ddif (advection and degenerate-diffusion) и Knob (advection) для C_{infl}^7
- Вывод: AMG хорошо работает и для адаптивных сеток тоже (!)



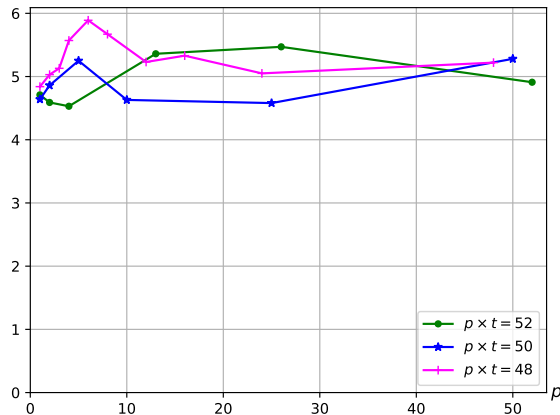
Время решения для гибридной MPI+OpenMP версии AMG

- P512 на 64 узлах кластера Ломоносова-2
- P256 на 1 узле кластера Сеченовского университета
- По оси абсцисс указано количество MPI процессов p

T (sec.)

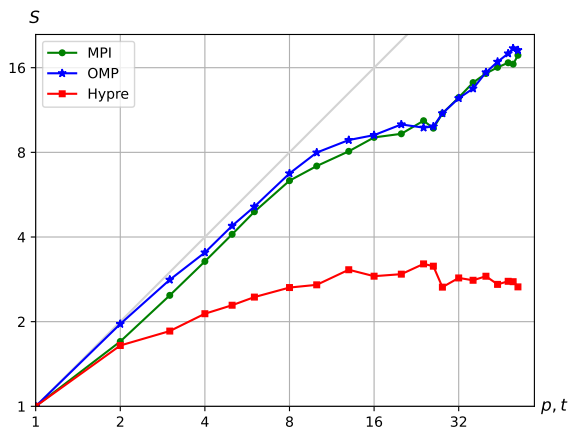
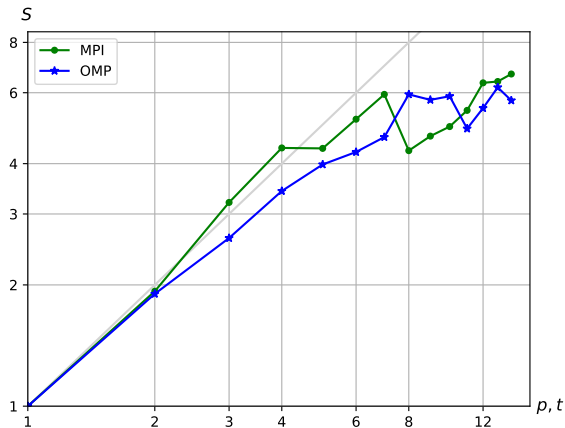


T (sec.)

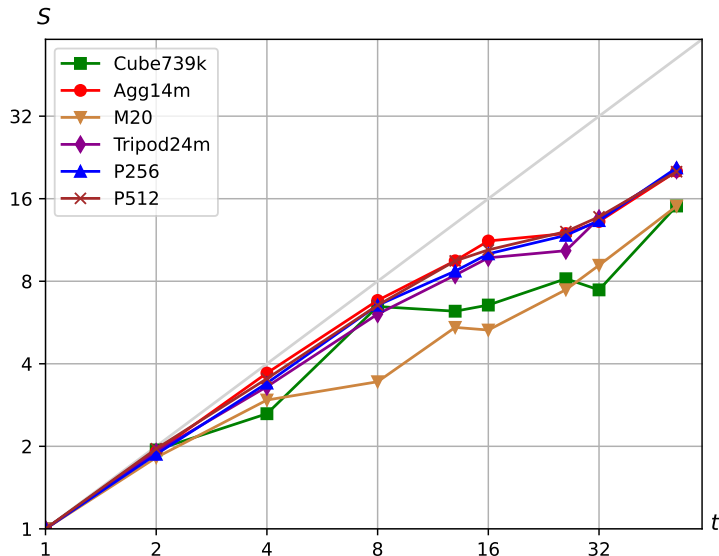


Сравнение масштабируемости MPI и OpenMP версий AMG

- P512 на 64 узлах кластера Ломоносова-2
- P256 на 1 узле кластера Сеченовского университета
- По оси абсцисс указано p или t , соответственно



Сильная масштабируемость (Speedup) для OpenMP



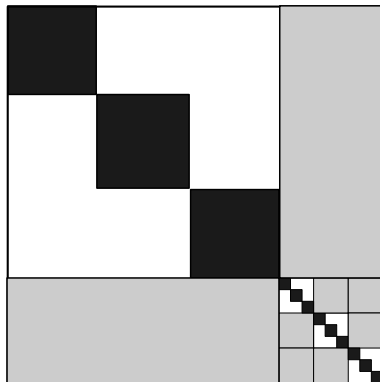
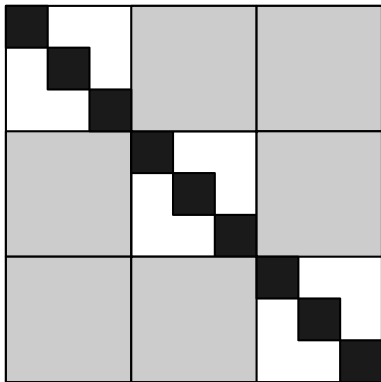
- $S_1 \approx 15$ для 2-х задач (1M и 20M)
- $S_1 \approx 20$ для 4-х задач (14M, 24M, 16M и 134M)
- 52 нити OpenMP

- Предложены, рассмотрены, реализованы, протестированы новые алгоритмы
- Для разных типов задач выбраны наиболее эффективные подходы
- В большинстве случаев эффективность выше чем у PETSc и HYPRE по:
 - времени решения (T)
 - памяти (OC)
 - масштабируемости (S)
- Новые OpenMP алгоритмы предстоит обобщить для распределенной памяти...

- I. N. Konshin, K. M. Terekhov, Reaching the scalability of a distributed AMG solver. *RJNAMM*, **41** (2026), No. 1, 45–67.
- I. N. Konshin, K. M. Terekhov, On improving the robustness and scalability of shared-memory AMG solvers for point-block problems. *RJNAMM*, V. 41 (2026), No. 2, 119-149.
- PETSc (the Portable, Extensible Toolkit for Scientific Computation).
<https://petsc.org/>
- HYPRE BoomerAMG. <https://github.com/hypre-space/hypre>,
<https://hypre.readthedocs.io/en/latest/solvers-boomeramg.html>

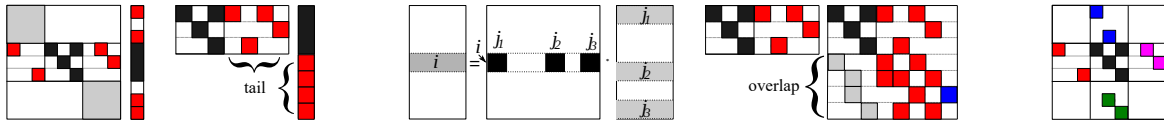
Алгоритмы упорядочивания

- Многоцветная перестановка (слева)
- Дважды окаймленная блочно-диагональная форма
- Черные квадраты — это разреженные диагональные блоки, серые блоки — разреженные подматрицы.



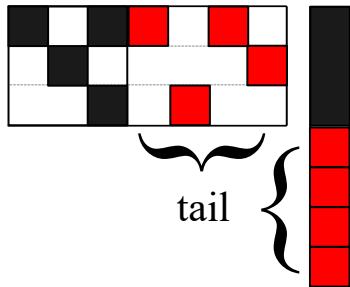
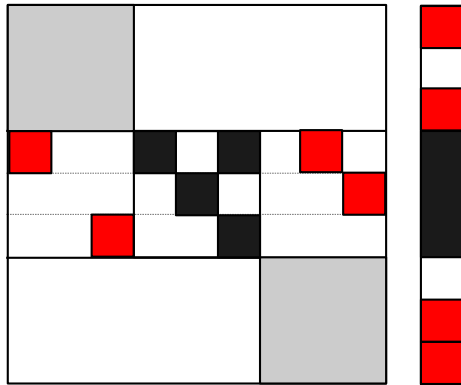
Операции с распределенными разреженными матрицами

- Умножение Ax : локальная перестановка нелокальных столбцов в «хвост».
- Произведение двух матриц AB по алгоритму Густавсона, когда «хвосты» строк B собираются и обмениваются.
- Транспонирование матрицы с обменом блоками в CSSR-формате.



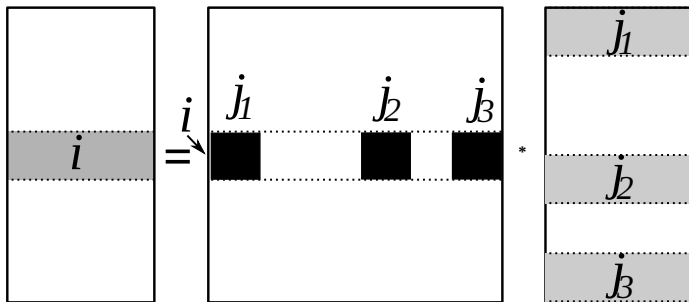
Глобальное и локальное представление матрицы и вектора

Распределенная операция преобразования разреженной матрицы в вектор на втором процессоре (слева) и ее локальное представление с нелокальными связями, перемещенное в конец (справа). Серые прямоугольники представляют нелокальные данные, черные прямоугольники — локальные ненулевые значения и локальные данные в векторе, красные прямоугольники — нелокальные ненулевые значения и нелокальные данные в векторе.



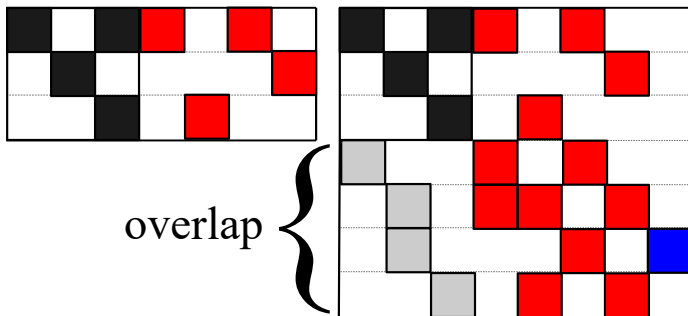
Вычисление i -й строки произведения двух матриц

Алгоритма Густавсона. Серые прямоугольники обозначают разреженные строки, а черные квадраты — ненулевые элементы в i -й строке.



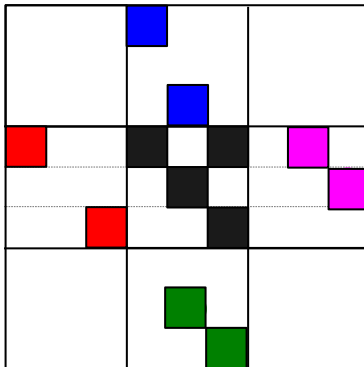
Произведение матрицы на себя: $A \cdot A$

Разреженная матрица, которая перекрывается при умножении сама на себя. Черные прямоугольники представляют локальные ненулевые значения и локальные данные, серые прямоугольники представляют ненулевые значения локального диапазона в области перекрытия, красные и синие прямоугольники представляют нелокальные ненулевые значения. Синий прямоугольник обозначает диапазон второго перекрывающегося слоя.



Транспонирование разреженной матрицы: A^T

Транспонирование распределенной разреженной матрицы на втором процессоре. Черные прямоугольники обозначают локальные ненулевые значения. Красные и фиолетовые прямоугольники обозначают ненулевые значения, которые должны быть отправлены процессорам один и три соответственно. Синие и зеленые прямоугольники обозначают ненулевые значения, которые должны быть получены от процессоров один и три соответственно.



Сокращение коммуникатора на грубых уровнях

- На грубых уровнях для маленьких матриц использовать много процессов неэффективно.
- После тройного умножения $B = RAP$ коммуникатор уменьшается, матрица перераспределяется.
- На итерациях вектора перераспределяются между коммуникаторами.
- Это повышает эффективность и экономит память.