

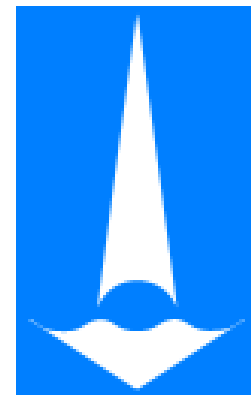
О реализации численных методов решения СЛАУ на основе гибридных моделей программирования

Б.И. Краснопольский

к.ф.-м.н.

***Старший научный сотрудник
Лаборатория общей аэродинамики
НИИ механики МГУ***

krasnopolsky@imec.msu.ru



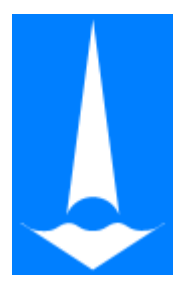


План доклада

Два направления:

- Библиотека численных методов решения разреженных систем линейных алгебраических уравнений
- Приложение для прямого численного моделирования турбулентных течений в областях сложной формы

***1. Библиотека численных методов
решения разреженных систем
линейных алгебраических уравнений***



1. Предназначение

- Подавляющее большинство задач механики сплошных сред в том или ином виде сводится к решению систем линейных алгебраических уравнений
- Одни из наиболее “тяжелых” - системы уравнений, полученные в результате разностной аппроксимации эллиптических уравнений в частных производных
- Эллиптические уравнения встречаются во многих областях физики:
 - ◆ Гидродинамика (расчет полей давления)
 - ◆ Электричество и магнетизм (распределения потенциалов)
 - ◆ Задачи напряженно-деформированного состояния материалов (прогиб мембран и оболочек)
- Время решения такой СЛАУ зачастую составляет более 90% по сравнению с остальными вычислениями при расчете нового шага по времени



1. Текущее состояние вопроса ...

Наиболее популярные библиотеки:

■ Trilinos

◆ Aztec/AztecOO

- × CG, CGS, BiCGStab, GMRES и пр.
- × ILU-предобуславливатели

◆ ML (> 3K procs)

- × AMG/SAMG

■ Hypre

- × CG, CGS, BiCGStab, GMRES и пр.

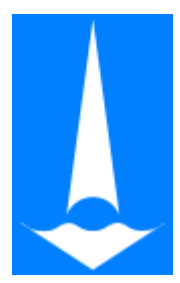
◆ BoomerAMG (~ 200K cores)

- × AMG

■ *LparSol (~ 10K cores)

- × CG, CGS, BiCGStab, GMRES и пр.
- × ILU-предобуславливатели
- × AMG





1. Hypre: On the Road to Exascale

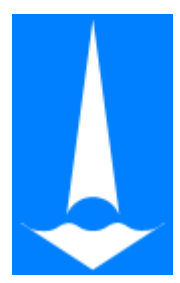
Позиционируется как один из инструментов для создания больших пользовательских вычислительных приложений в **Exascale**-перспективе



Scaling Algebraic Multigrid Solvers: On the Road to Exascale

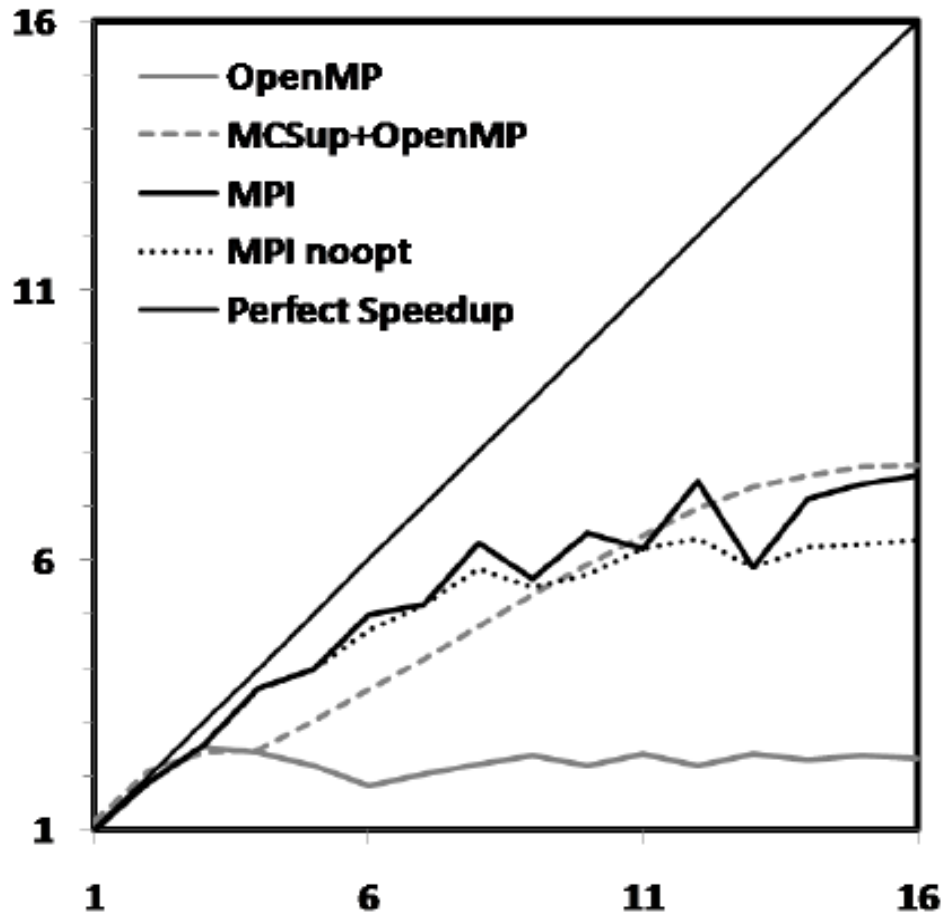
Allison H. Baker, Robert D. Falgout, Todd Gamblin, Tzanio V. Kolev, Martin Schulz and Ulrike Meier Yang

* A.H. Baker, R.D. Falgout, T. Gamblin, Tz.V. Kolev, M. Schulz, and U.M. Yang. Scaling Algebraic Multigrid Solvers: On the Road to Exascale. Proceedings of Competence in High Performance Computing, CiHPC 2010.



1. *Hypr*: MPI+OpenMP «Почти» open-source

Масштабируемость внутри одного узла



Произведение разреженной матрицы на вектор:

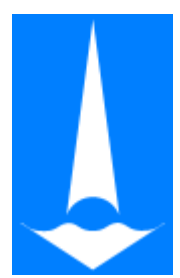
◆ Использование OpenMP модели приводит к ускорению в 2-2.5 раза!

◆ **MCSup**: Multi-Core Support library, API для управления выделением памяти между процессорами и привязки процессов

◆ **MCSup** не распространяется с *hypr*!

* 4 x Quad-core AMD Opteron processors (4 NUMA nodes)

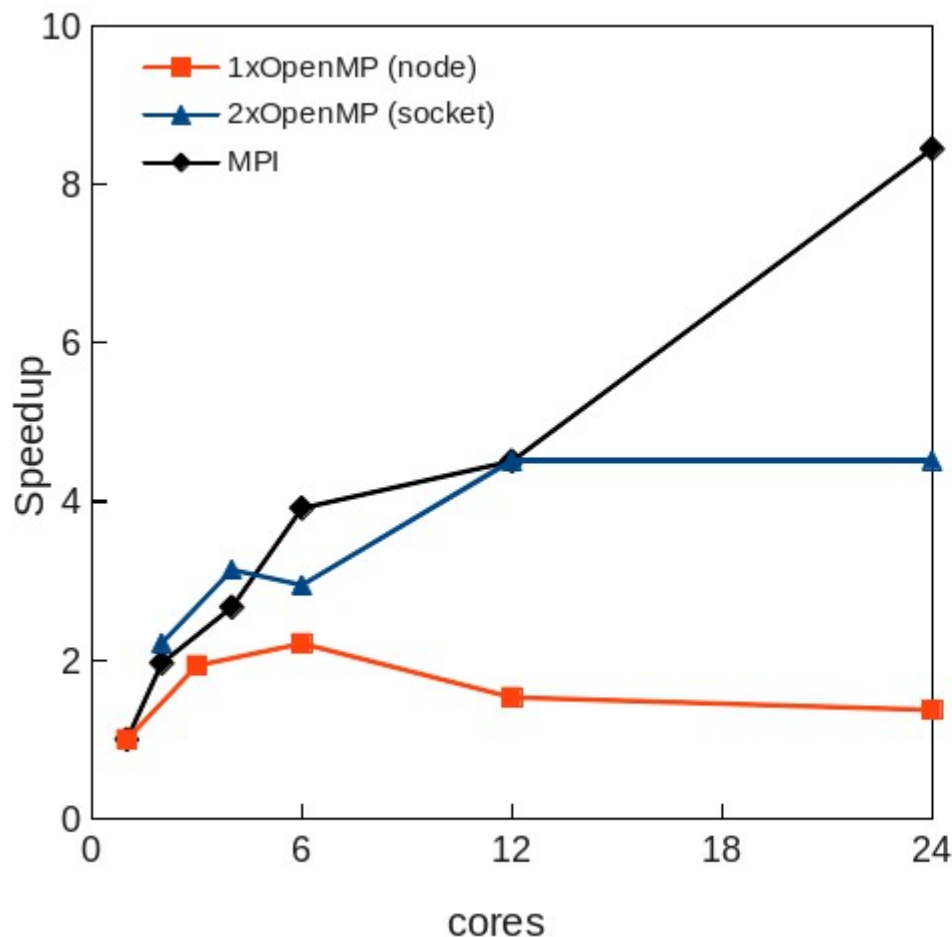
* A.H. Baker, R.D. Falgout, T. Gamblin, Tz.V. Kolev, M. Schulz, and U.M. Yang. Scaling Algebraic Multigrid Solvers: On the Road to Exascale. Proceedings of *Competence in High Performance Computing*, CiHPC 2010.



1. Нурге: MPI+OpenMP Зилант и Ломоносов

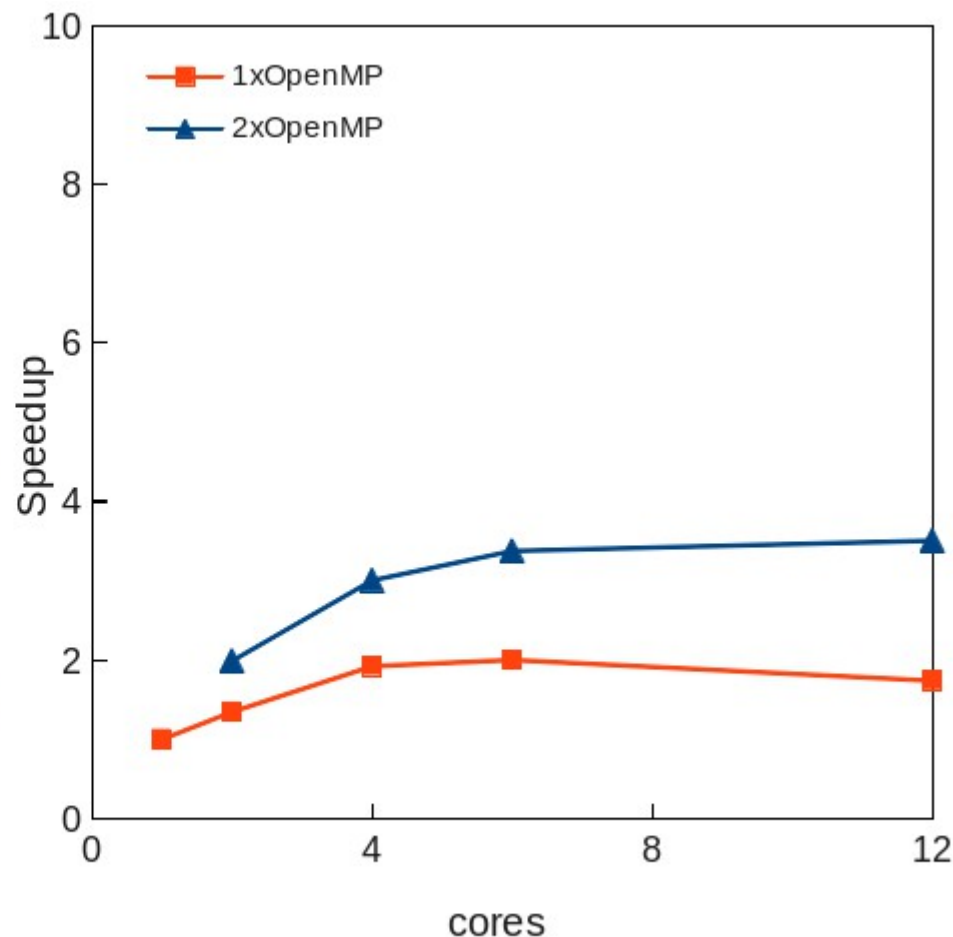
Масштабируемость внутри одного узла

3.3M, BiCGStab

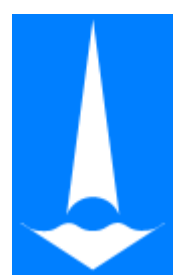


2 x 12-core AMD Opteron
4 NUMA nodes

1M, BiCGStab+AMG

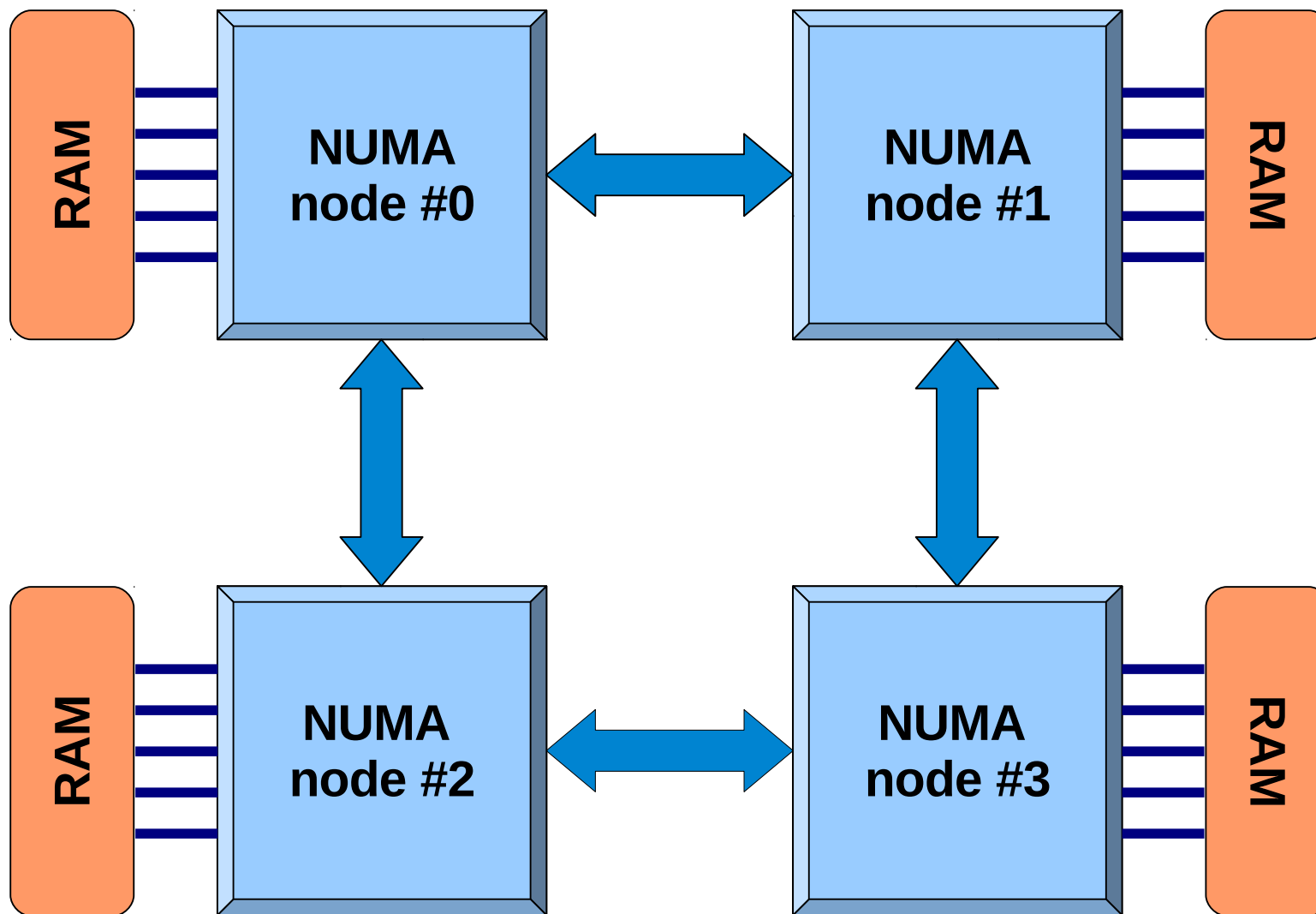


2 x 6-core Intel Nehalem
2 NUMA nodes



1. NUMA архитектура

Non-Uniform Memory Access





1. *Hyprе*, или можно лучше?..

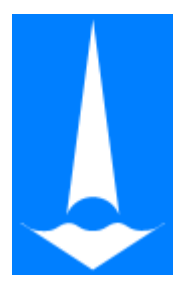
*Можно ли добиться повышения эффективности методов решения больших сильно-разреженных систем линейных алгебраических уравнений (итерационные методы подпространства Крылова, многосеточные методы) по сравнению с **hyprе**?..*



1. Направления поиска

Возможные пути повышения эффективности методов:

- Модификация численных методов
- Использование других гибридных моделей программирования
- Многомерное разбиение матриц
- Балансировка вычислений на всех уровнях вложенности
- Уменьшение количества процессов на нижних уровнях вложенности
- ...



1. Модификация численных методов

Reordered BiCGStab

Типы вычислительных операции метода:

$$y = Ax, \quad y = K^{-1}x, \quad z = \alpha x + \beta y, \quad \alpha = (x, y)$$

Classical BiCGStab

1. $\hat{v} = K^{-1}v$
2. $t = A\hat{v}$
3. $\alpha = (t, x)$
4. $s = y + \alpha z$

Reordered BiCGStab

1. $t = A\hat{v}$
2. $\alpha = (t, x)$
3. $p = K^{-1}r$
4. $s = y + \alpha z$

MPI_Allreduce — неблокирующий аналог глобальной редукции

* B. Krasnopol'sky. The Reordered BiCGStab Method For Distributed Memory Computer Systems // *Procedia Computer Science*, 2010, v. 1, pp. 213-218.



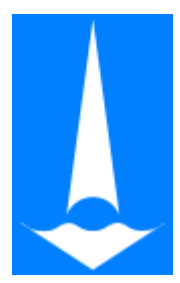
1. MPI_IAllreduce

По аналогии с *Mvapich*:

- ◆ Редукция между процессами внутри узла: через общую память
- ◆ Редукция между узлами: коммуникации «точка-точка» по биномиальному дереву

Прототипы функций (в стадии реализации):

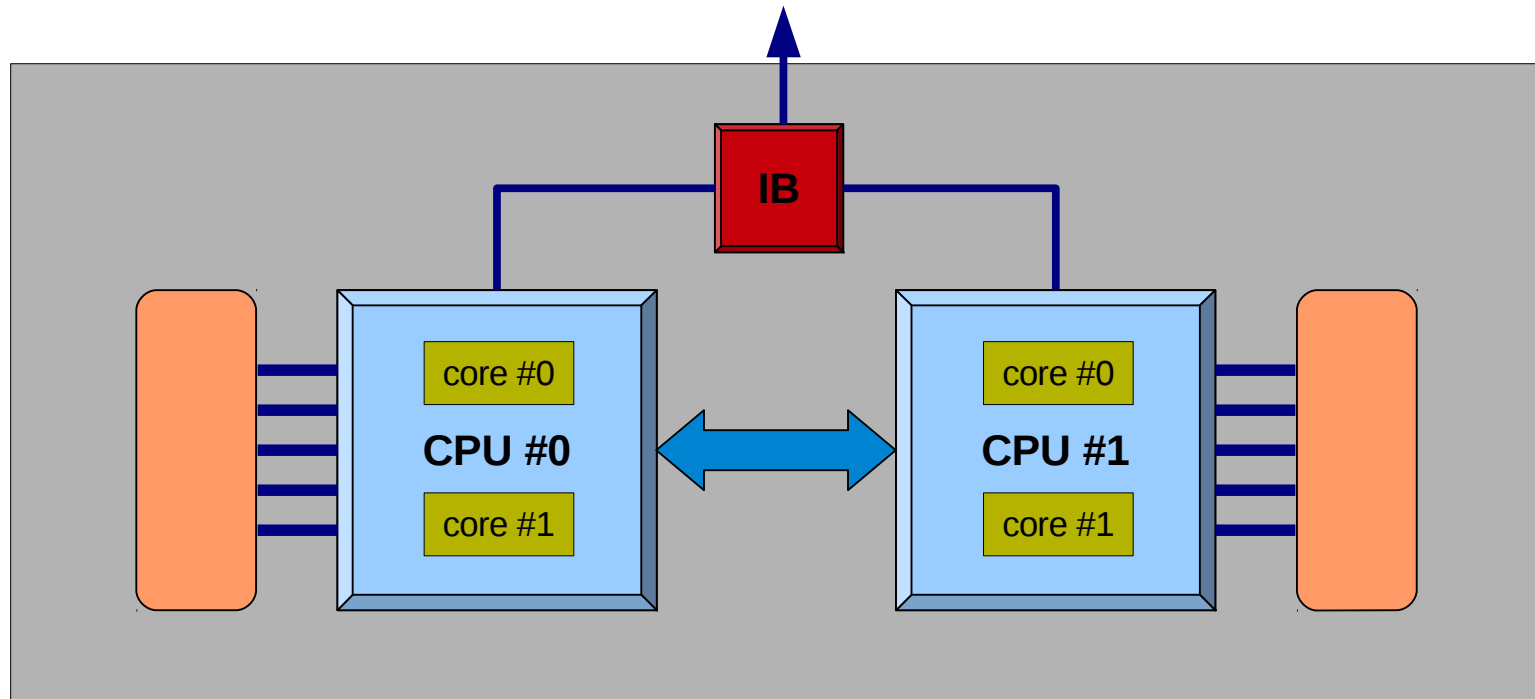
<i>MPIShM_IAllreduce_init (...)</i>	Инициализация
<i>MPIShM_IAllreduce_shm_op (...)</i>	Локальная редукция
<i>MPIShM_IAllreduce_shm_finalize (...)</i>	Завершение локальной редукции, запуск глобальной операции
<i>MPIShM_IAllreduce_process (...)</i>	Обработка завершившихся обменов
<i>MPIShM_IAllreduce_finalize (...)</i>	Завершение операции

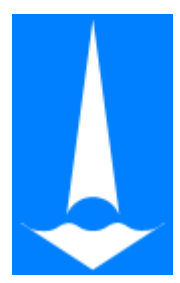


1. Гибридная модель MPI+ShM

MPI + Posix Shared Memory

- Более низкоуровневая модель по сравнению с MPI+OpenMP
- Простой и прозрачный способ распределения и привязки процессов между ядрами внутри узла

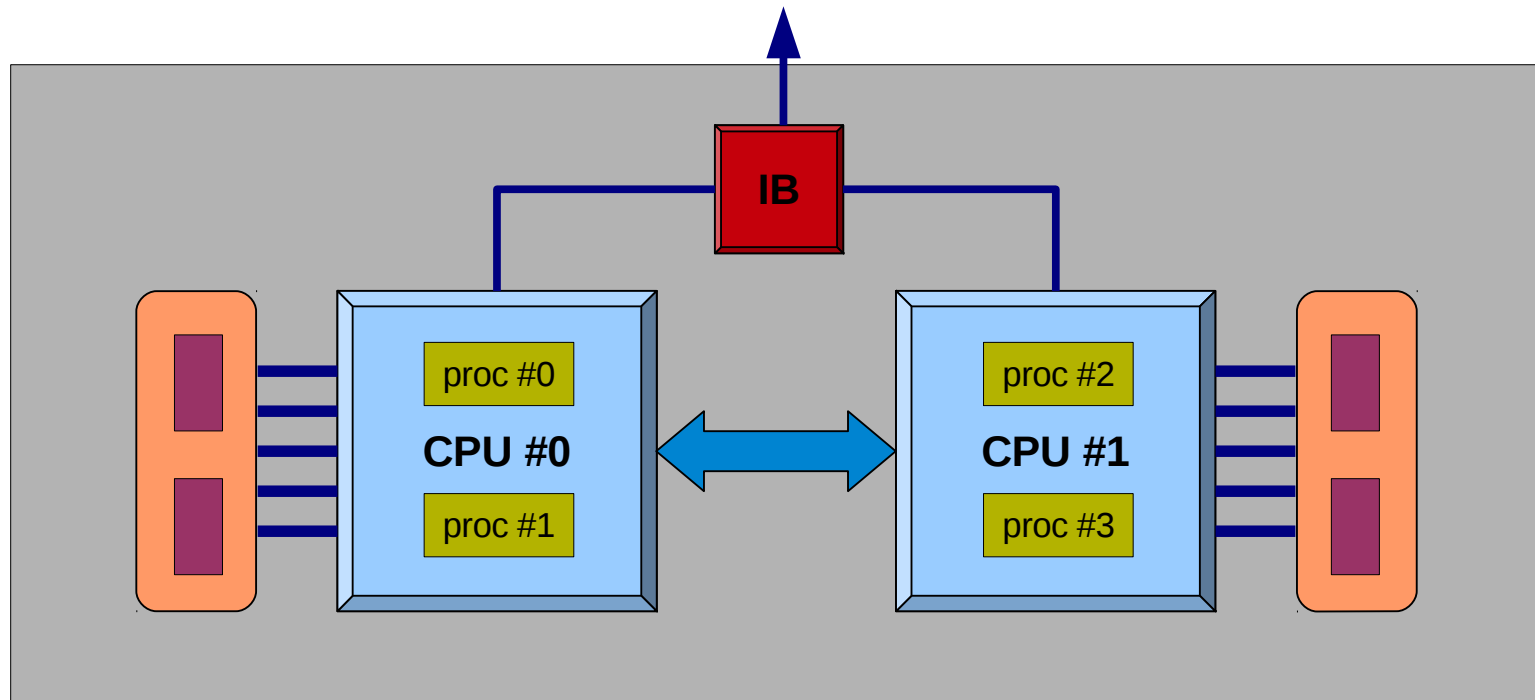


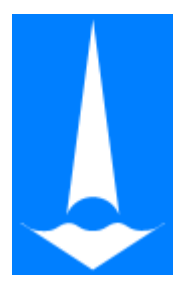


1. Гибридная модель MPI+ShM

MPI + Posix Shared Memory

- Более низкоуровневая модель по сравнению с MPI+OpenMP
- Простой и прозрачный способ распределения и привязки процессов между ядрами внутри узла
 - ◆ Запуск обычной MPI программы

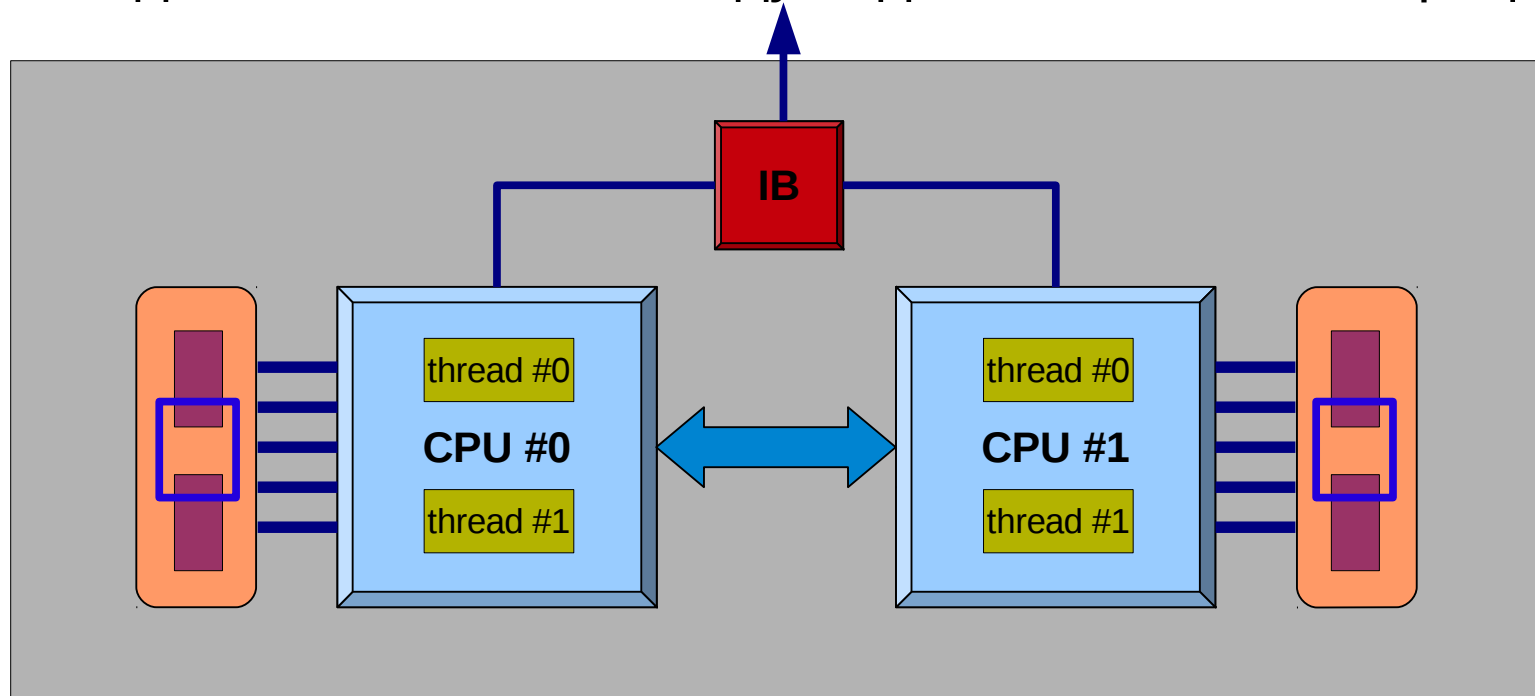




1. Гибридная модель MPI+ShM

MPI + Posix Shared Memory

- Более низкоуровневая модель по сравнению с MPI+OpenMP
- Простой и прозрачный способ распределения и привязки процессов между ядрами внутри узла
 - ◆ Запуск обычной MPI программы
 - ◆ «Объединение» памяти между подмножеством MPI-процессов





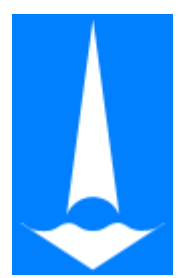
1. Гибридная модель MPI+ShM

Умножение матрицы на вектор

Простейший вариант умножения матрицы на вектор: $y = Ax$

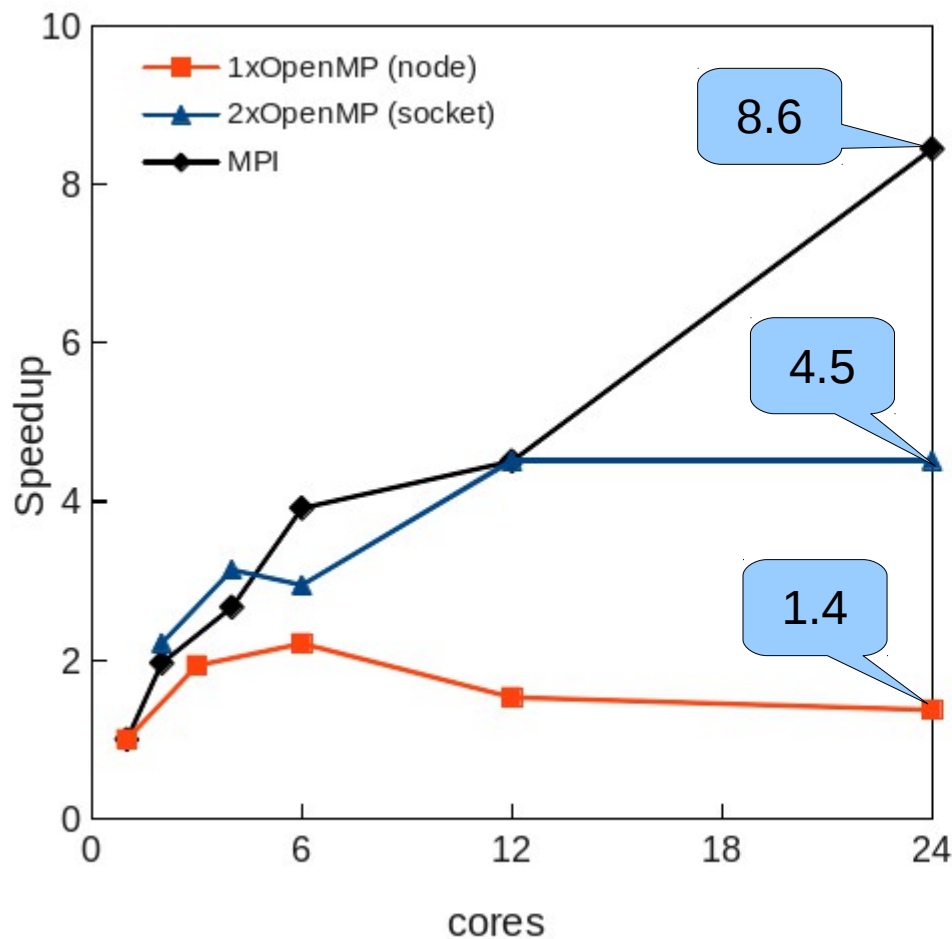
- Вектор x расположен в общей памяти;
- Вектор x целиком доступен всем процессам, но принадлежит одному процессу (выделен в одном NUMA-узле)

* Априори **неэффективно** для NUMA архитектуры, но дает оценку нижнего предела эффективности гибридной модели

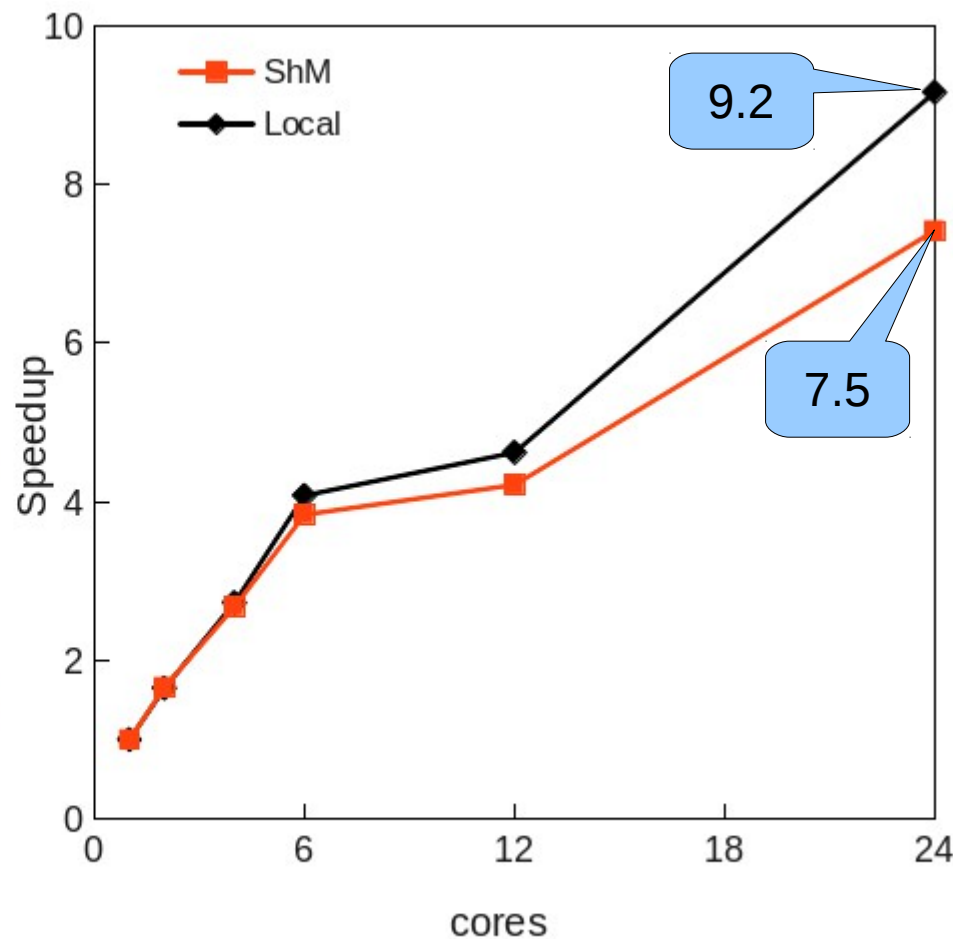


1. Гибридная модель MPI+ShM BiCGStab, результаты, Зилант

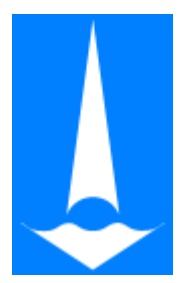
Размер матрицы 3.3М неизвестных, до 7 ненулевых элементов в каждой строке



hypre



SparseLinSol



1. Многомерное разбиение матриц

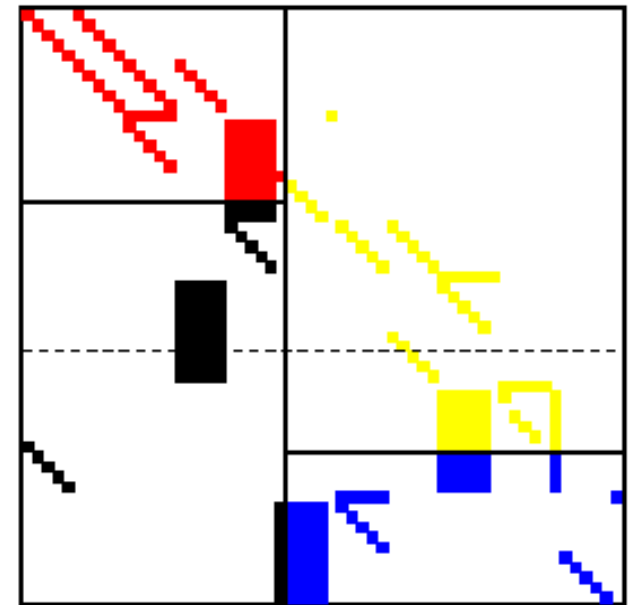
«Стандартный» подход: построчное разбиение матрицы между вычислительными процессами

Альтернативы:

- Одномерное разбиение по столбцам
- Двумерное разбиение матрицы?..

Алгоритмы поиска оптимального разбиения графов и гиперграфов:

- ***ParMetis***
- ***PT-Scotch***
- ***Zoltan***





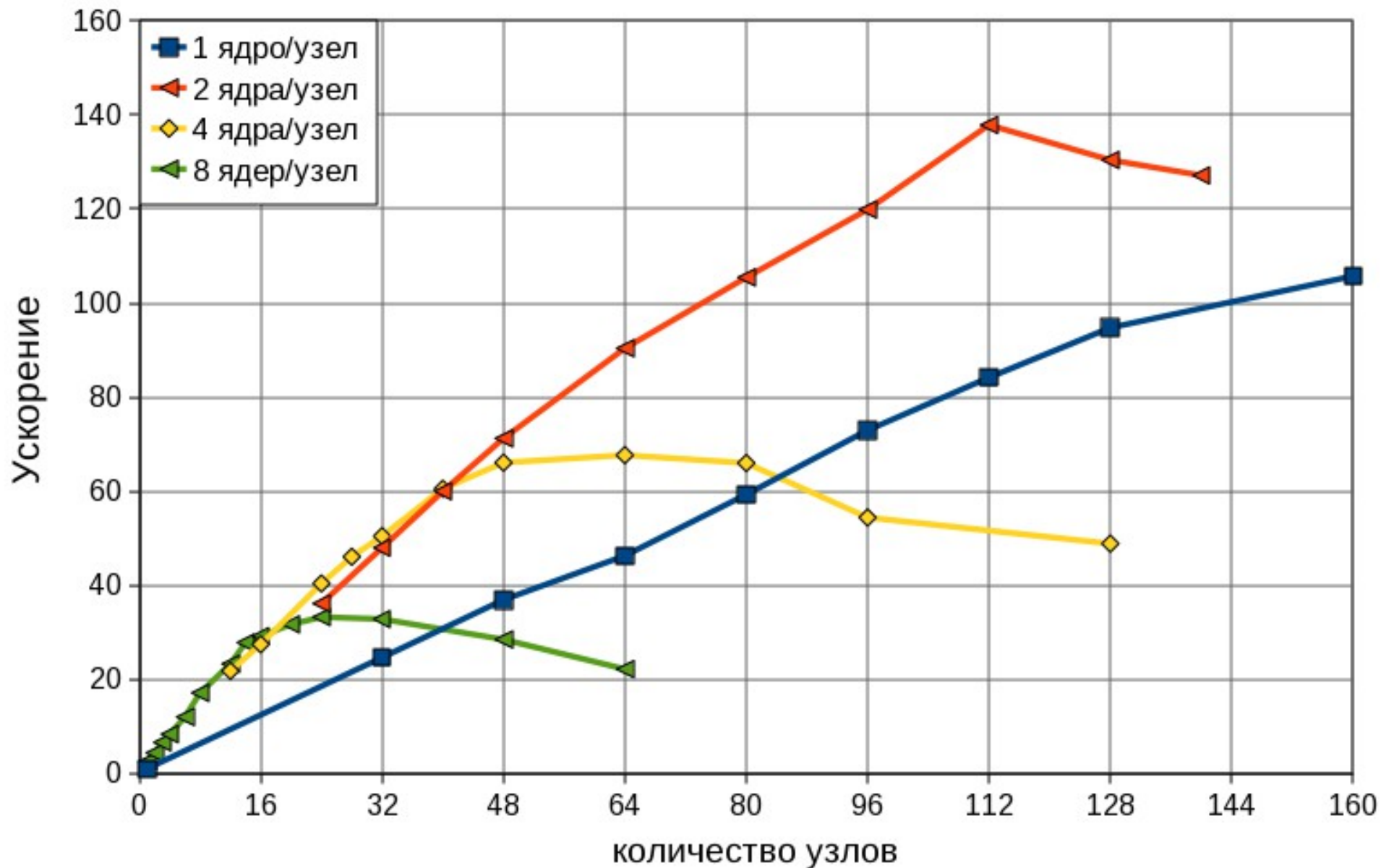
1.*Тестовые задачи

- Задача Неймана для уравнения Пуассона в кубической области на равномерной сетке (7-точечный шаблон пространственной аппроксимации)
- Фиксированное количество итераций метода (BiCGStab+AMG)
- Ускорение определено как:

$$P_N = \frac{T_1}{T_N}$$

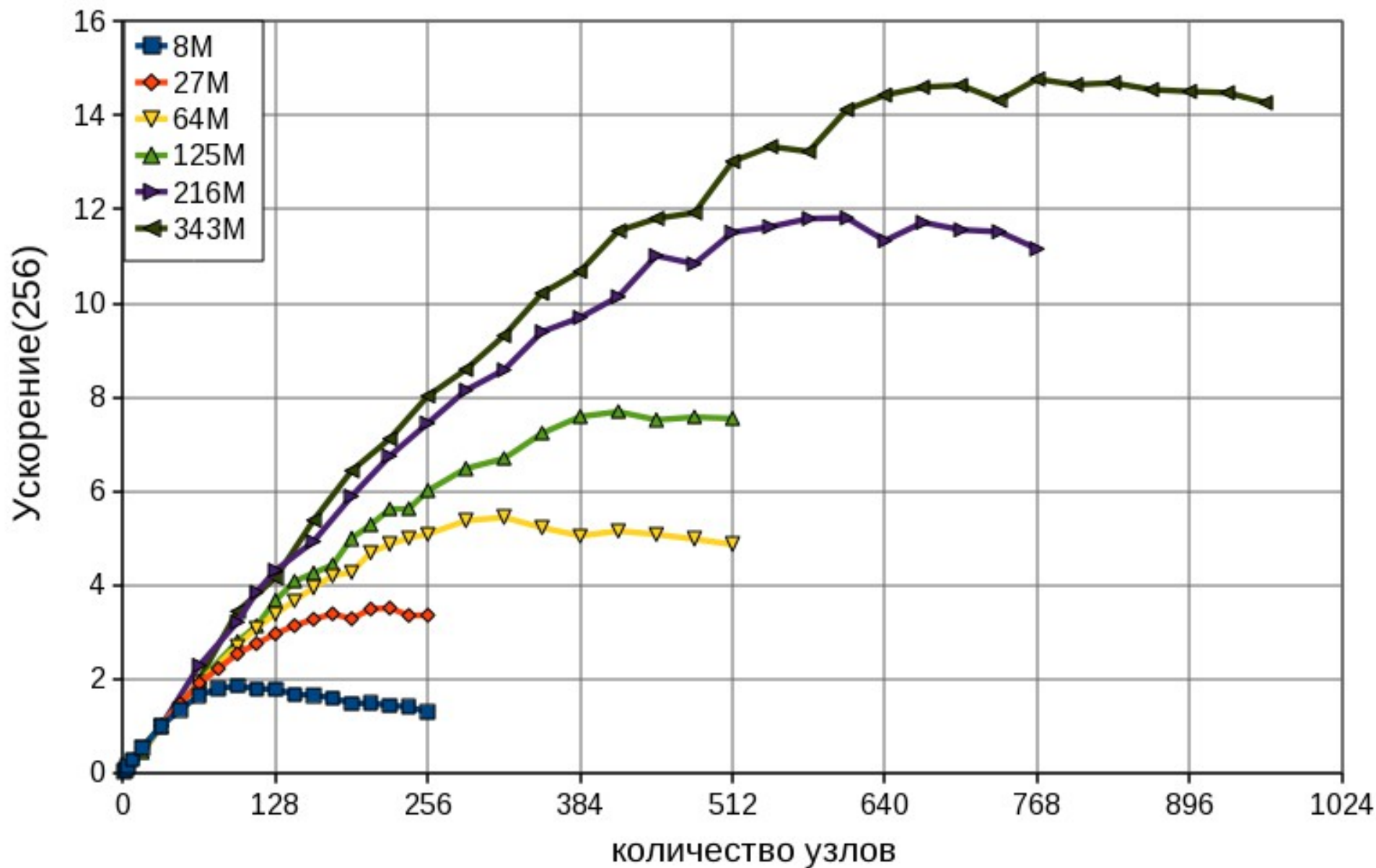


1.*Результаты масштабируемости (8М) Чебышёв, MPI





1.*Результаты масштабируемости Ломоносов, MPI+ShM

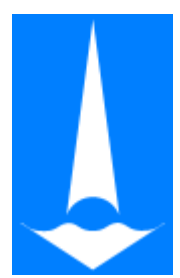




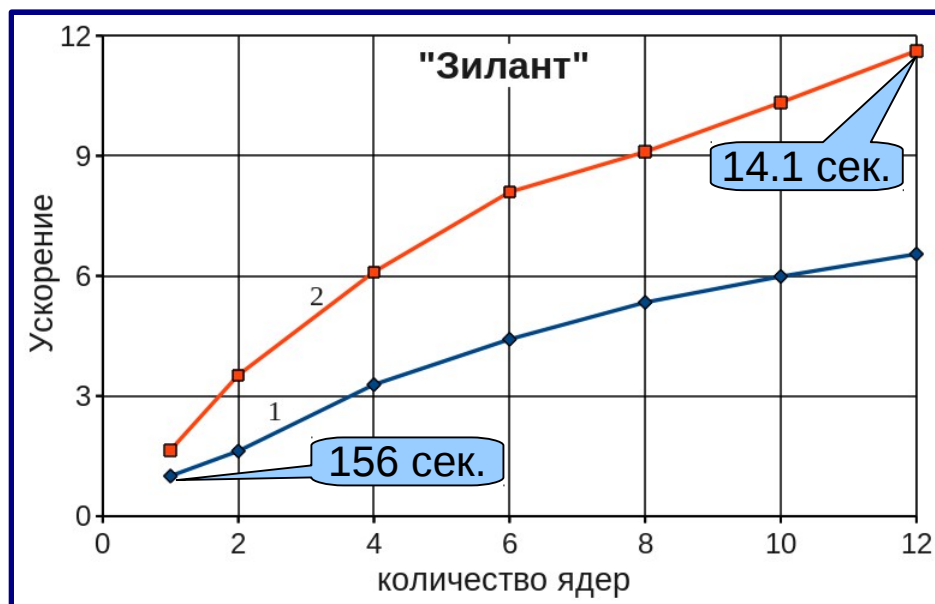
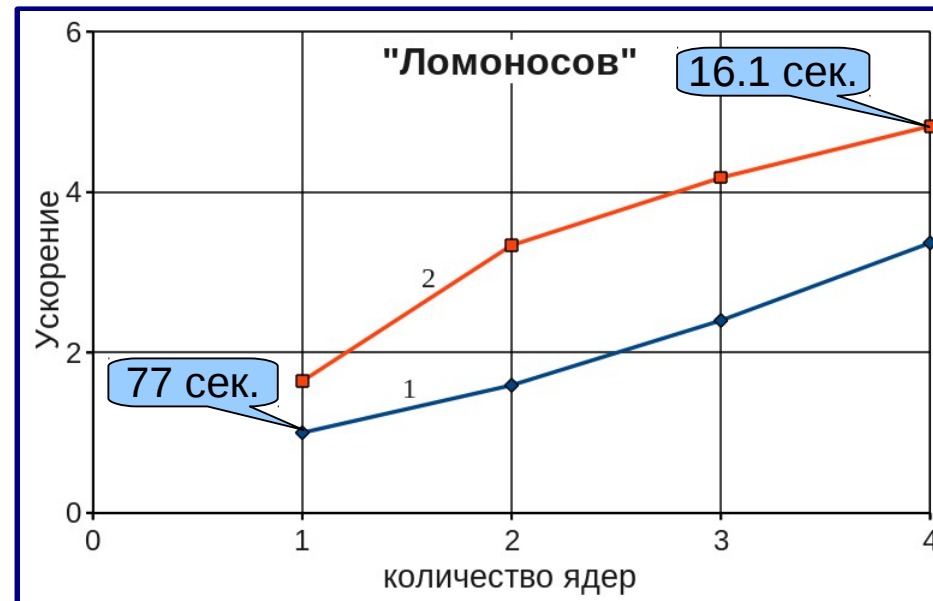
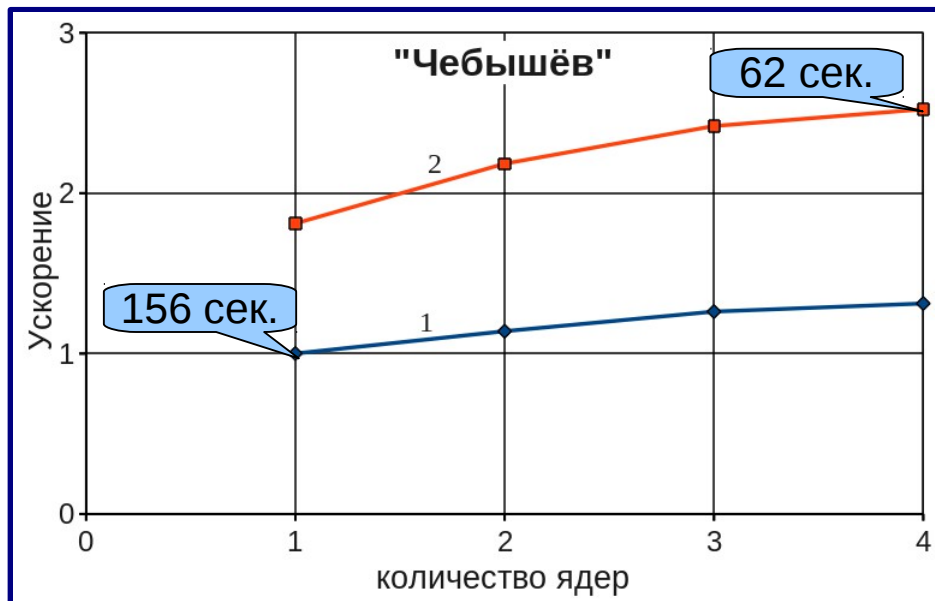
1.*Результаты масштабирования Ломоносов, MPI+ShM

Пиковые ускорения для различных тестовых матриц:

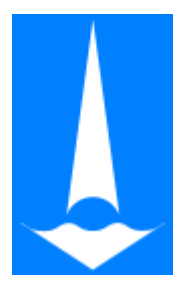
Размер матрицы, млн. строк	Оптимальное количество узлов	Пиковое ускорение
8	96	178
27	224	337
64	320	522
125	416	738
216	608	1134
343	768	1417



1.*Внутриузловая масштабируемость (8M) MPI



1 – один процессор
2 – два процессора



1.*Внутриузловая масштабируемость (8M) MPI+ShM

Вычислительная система	Размер блоков							
	1	2	3	4	6	8	12	24
“Чебышёв”	62	61.74	-	61.25	-	60.43	-	-
“Ломоносов”	16.13	17.04	-	22.6	-	21.1	-	-
“Зилант”	14.13	12.92	13.06	-	13.89	-	23.88	49.19

- Используются все имеющиеся ядра на узле
- Размер блока – количество “агрегированных” MPI-процессов

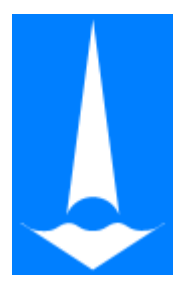


1. Вместо заключения

*Можно ли добиться повышения эффективности методов решения больших сильно-разреженных систем линейных алгебраических уравнений (итерационные методы подпространства Крылова, многосеточные методы) по сравнению с **hypr**?..*

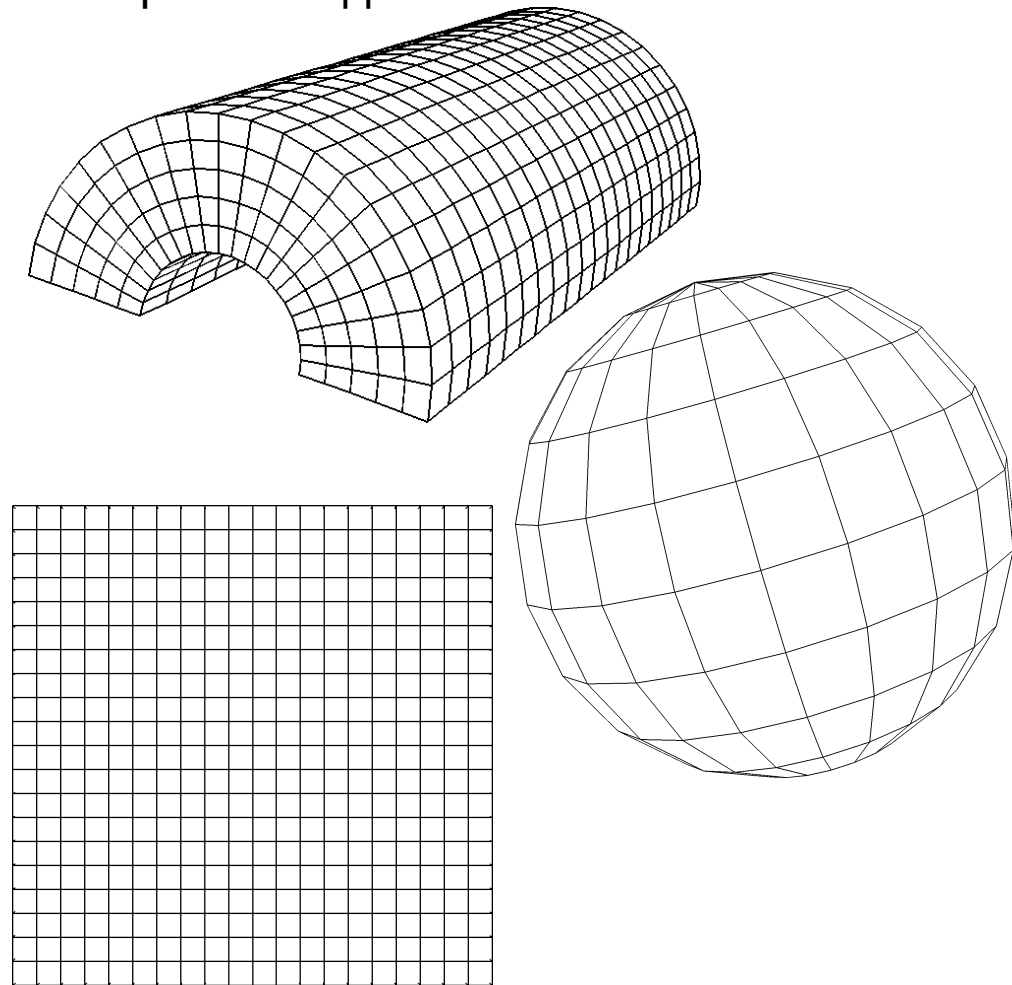
ДА!

***2. Приложение для прямого
численного моделирования
турбулентных течений в областях
сложной формы***

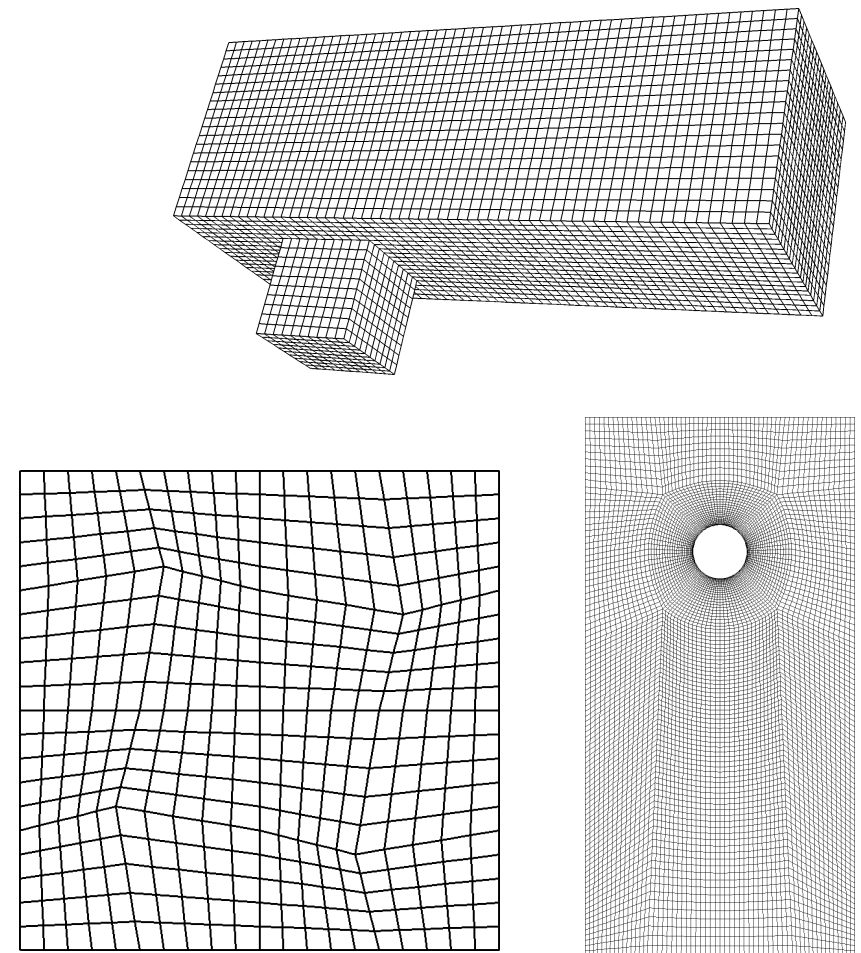


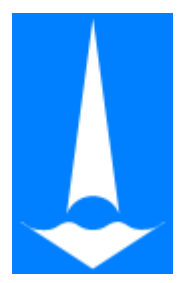
2. «Простые» и «сложные» области

«Простые» области – задачи, где для решения уравнения Пуассона для давления возможно применение прямых быстрых методов



«Сложные» области – задачи, где для решения уравнения Пуассона для давления необходимо использование методов общего вида





2. Система уравнений и граничные условия

Рассматривается течение вязкой несжимаемой жидкости

Уравнение неразрывности:
$$\frac{\partial u_j}{\partial x_j} = 0$$

Уравнения Навье-Стокса:
$$\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} = \frac{1}{\text{Re}} \frac{\partial^2 u_i}{\partial x_j \partial x_j}$$

Число Рейнольдса:
$$\text{Re} = \frac{UD}{\nu}$$

Граничные условия:

- ♦ *распределение скоростей по границе,*
- ♦ *прилипание,*
- ♦ *проскальзывание,*
- ♦ *периодичность,*
- ♦ *распределение давления по границе,*
- ♦ **«сопряженные» расчеты***



2. Вычислительный алгоритм

Расчет шага по времени:

1. Расчет предварительного поля скоростей в центрах ячеек:

схема Адамса-Башфорта:
$$\frac{\hat{u}_i - u_i^k}{\Delta t} = 3 \left[(NL + VISC)^k - (NL + VISC)^{k-1} \right] / 2$$

2. Расчет предварительных скоростей на гранях ячеек

$$\hat{U} = \left(\frac{\hat{u}_{i\ cv1} + \hat{u}_{i\ cv2}}{2} \right) n_i$$

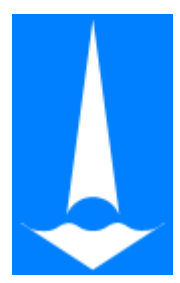
3. Формулировка уравнения для давления относительно скоростей на гранях ячеек, расчет скоростей на гранях ячеек

$$\frac{U_n - \hat{U}}{\Delta t} = - \frac{\partial p}{\partial n}$$

4. Расчет поля скоростей в центрах ячеек

$$\frac{u_i^{k+1} - \hat{u}_i}{\Delta t} = - \frac{\partial p}{\partial x_i}$$

* Mahesh K., Constantinescu G., Moin P. A numerical method for large-eddy simulation in complex geometries // *Journal of computational physics*. 2004. V. 197. Pp. 215-240.



2. Вычислительный алгоритм

Аппроксимация нелинейных членов

$$NL = \frac{1}{V_{cv}} \sum_{f \text{ of } cv} U_n^f \left(\frac{u_{i cv} + u_{i nbr}}{2} \right) A_f$$

В общем случае обеспечивает первый порядок аппроксимации, **НО сохраняет дискретную кинетическую энергию**

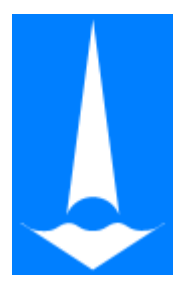
Аппроксимация градиента давления на гранях ячеек

$$\frac{\partial p}{\partial n_f} = \frac{p_{cv_1} - p_{cv_2}}{d_f}$$

Аппроксимация градиента давления в центрах ячеек

$$S = \sum_{f \text{ of } cv} \left(\frac{\partial p}{\partial x_i} n_i^f - \frac{\partial p}{\partial n_f} \right) A_f$$

Минимизация функционала S для каждой из ячеек относительно трех переменных

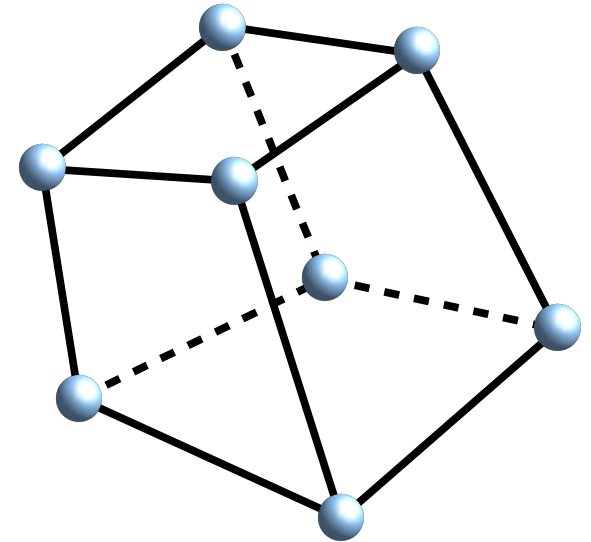


2. Особенности реализации

Неструктурированные сетки: ячейка описывается **только** номерами восьми образующих вершин.

Вся информация о сетке хранится в двух списках:

1. **Список вершин**, содержит номер и координаты вершины
2. **Список ячеек**, содержит номер ячейки и номера восьми образующих ячейку вершин



При построении аппроксимации уравнений используется еще один вспомогательный список:

3. **Список граней**, содержит номер грани и номера образующих эту грань ячеек

Такая схема представления расчетных сеток обеспечивает минимальное дублирование данных между вычислительными процессами и хорошую масштабируемость приложения по памяти.

2. Входные и выходные данные

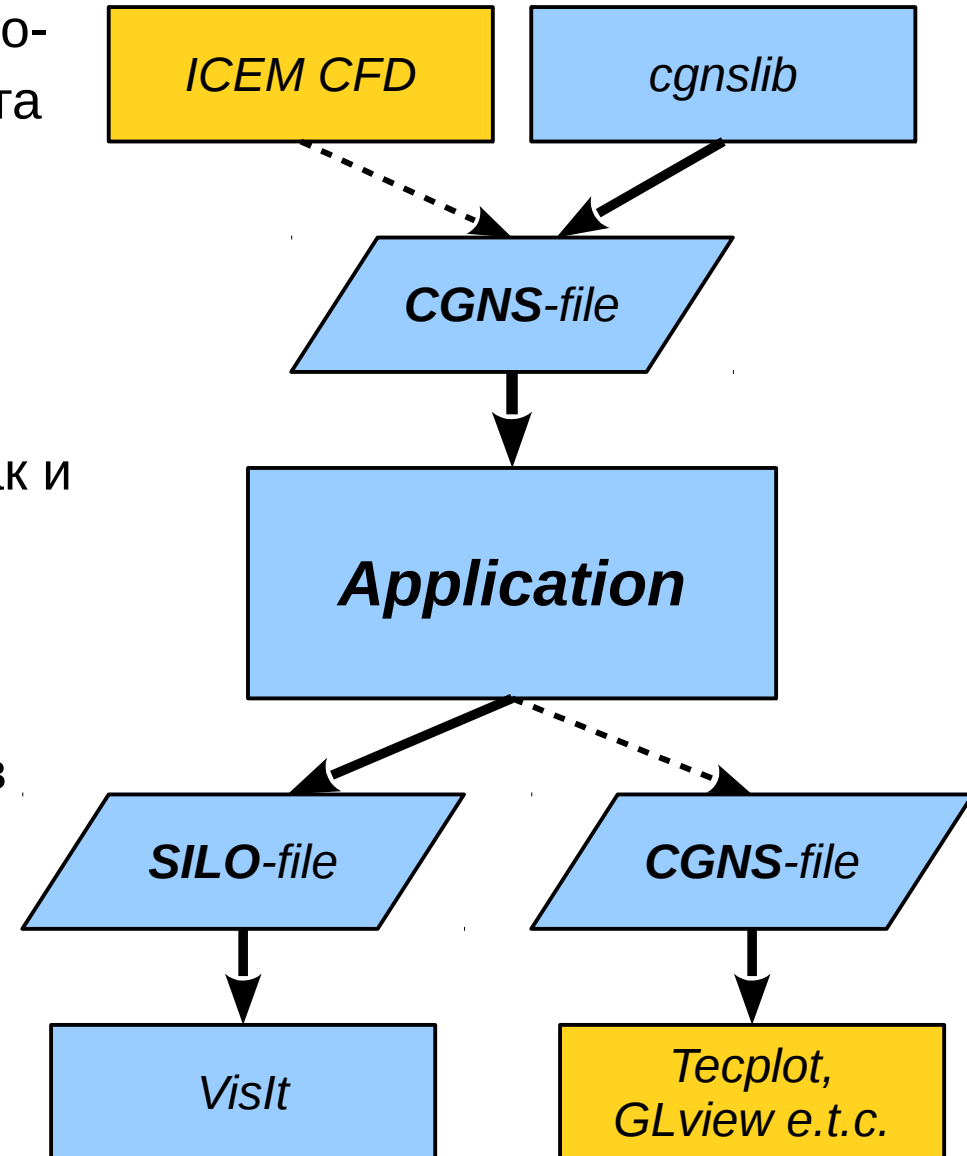
Интерфейс с существующими специализированными пакетами для пре/пост-процессинга

♦ Входные данные: **CGNS** формат

- ♦ Информация о расчетной сетке и граничных условиях
- ♦ Файл может быть создан как из коммерческих сеточных генераторов, так и с помощью специализированной open-source библиотеки

♦ Выходные данные: **SILO** формат

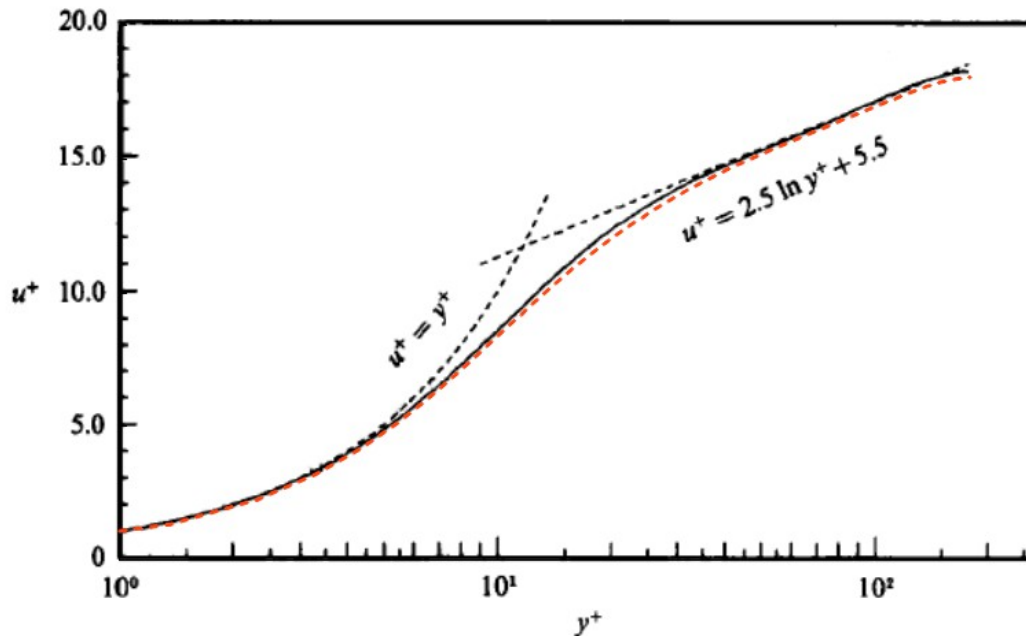
- ♦ Обеспечивает простой импорт данных в пакет визуализации **VisIt**
- ♦ Ориентирован на обработку больших объемов данных
- ♦ Удобен для записи данных — данные вычислительных процессов могут быть записаны на диск независимо



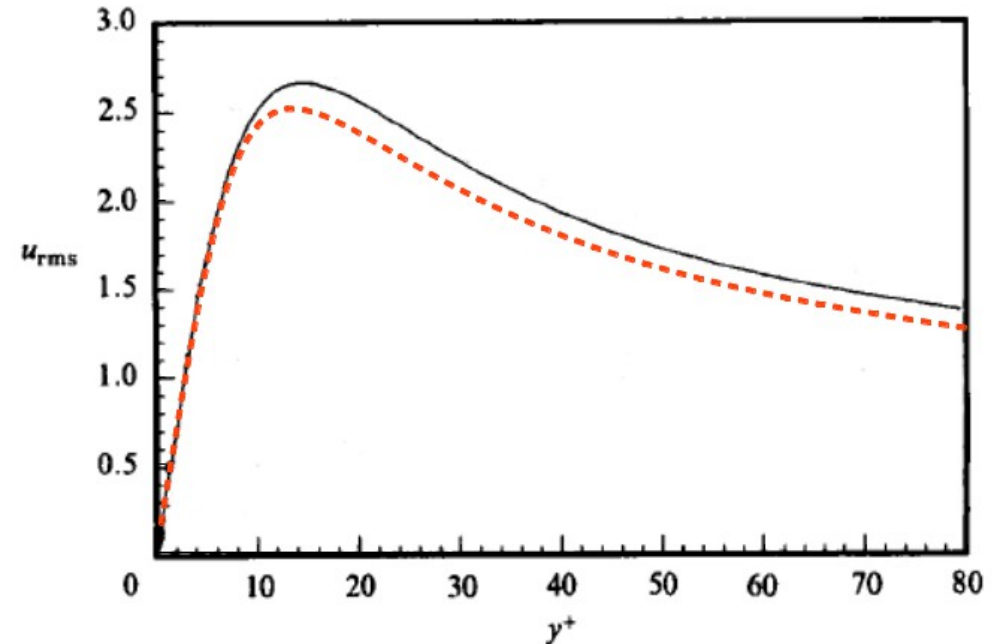


2. Первичная верификация турбулентное течение в плоском канале

$$Re_\tau = 180 \quad 128^3 \text{ ячеек}$$



Осредненный профиль продольной
компоненты скорости



Распределение среднеквадратичных
пульсаций продольной компоненты
скорости, в сравнении

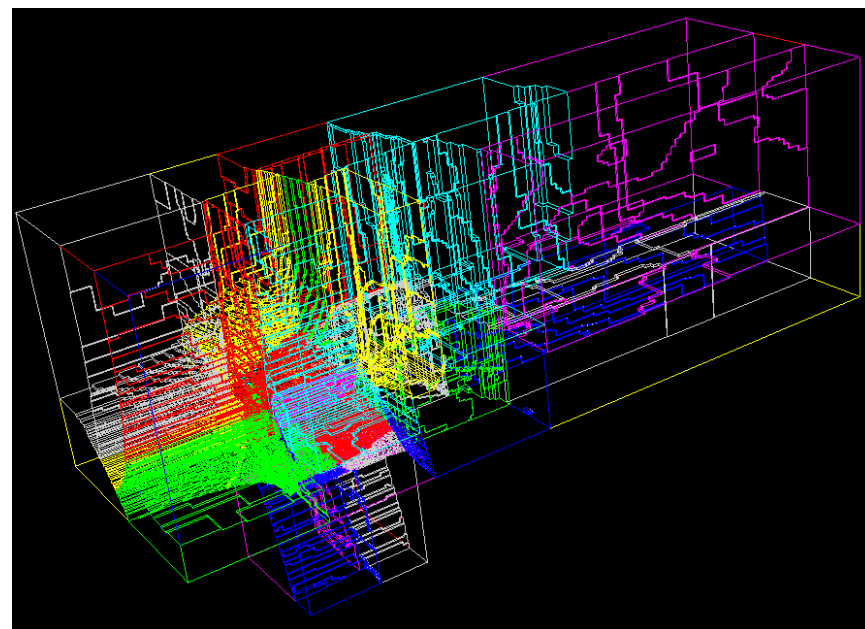
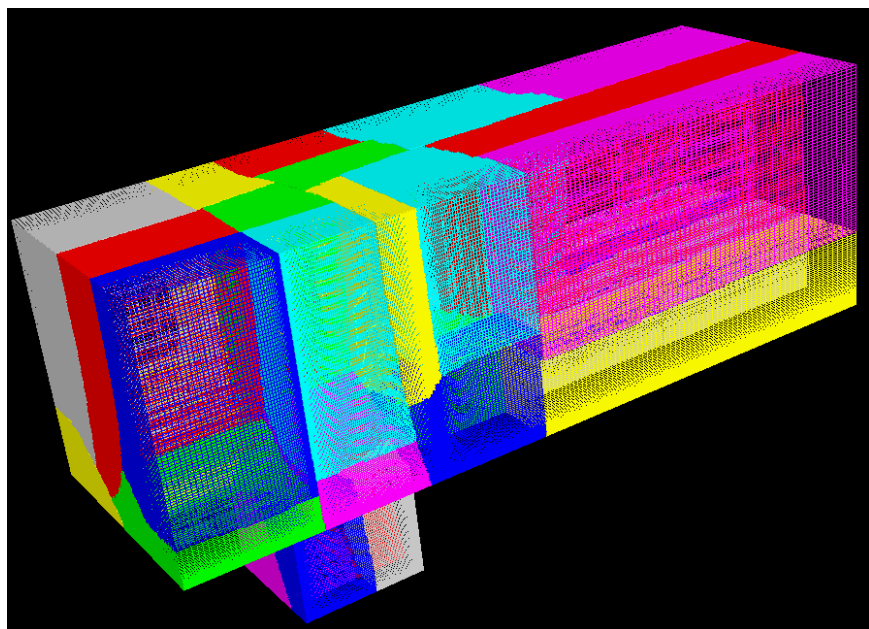
В сравнении с: Kim J., Moin P., Mozer R. Turbulence statistics in fully developed channel flow at low Reynolds number // Journal of Fluid Mechanics, 1987, V. 177, P. 133-166.



2. Тестовая задача

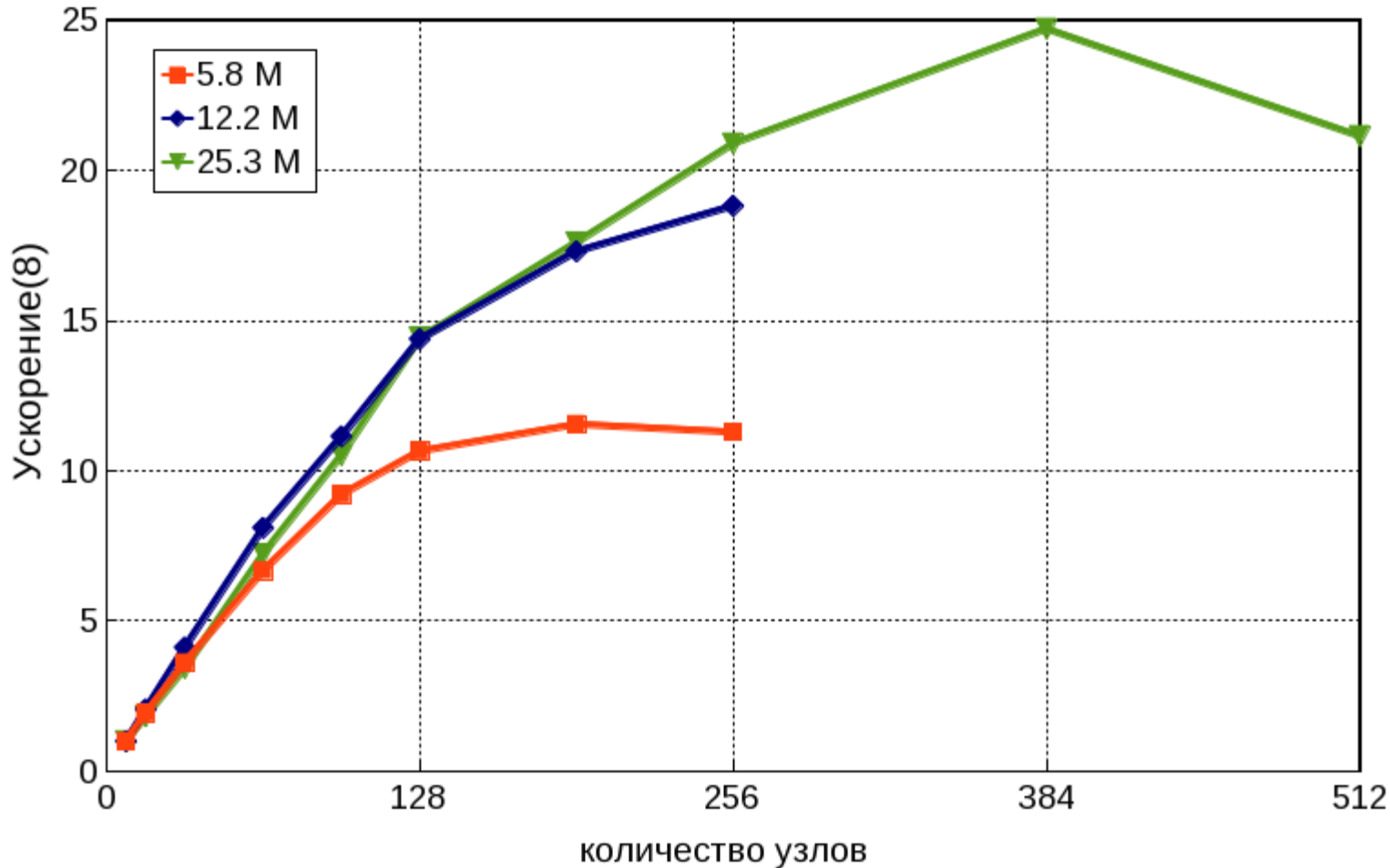
Первичные результаты масштабируемости получены для трех тестовых задач обтекания трехмерной прямоугольной каверны

	Количество ячеек, млн.	Количество вершин, млн	$h_{min}, \cdot 10^4$	h_{max}/h_{min}
mdl1	5.8	5.9	6.8	76.3
mdl2	12.2	12.4	5	40
mdl3	25.3	25.6	6.2	20.8





2. Результаты масштабируемости Ломоносов, MPI



Размер задачи, М	Ускорение
5.8	85
12.2	140
25.3	180

- ◆ Расчет 100 шагов по времени
- ◆ Использовано только одно ядро на узле

Размер задачи, М	Время расчета на 8 узлах, сек	Минимальное время, сек
5.8	805	69.7
12.2	2056	109.2
25.3	3827	154.9

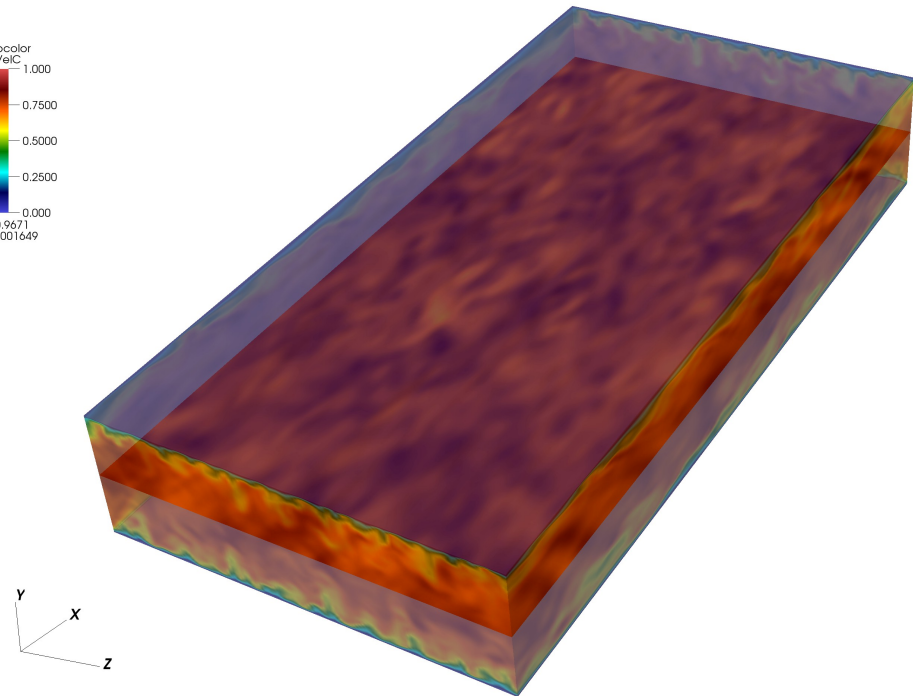
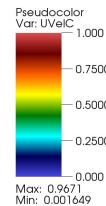


2. Заключение

* Создан прототип приложения для расчета турбулентных течений в областях сложной формы. Заложенные в основу методы и алгоритмы обеспечивают хорошие характеристики ускорения времени расчета задачи и масштабируемости по памяти.

* Получены первичные данные верификации алгоритма, свидетельствующие об удовлетворительной точности результатов расчетов.

* Создание интерфейса для проведения «сопряженных» расчетов обеспечит возможность расчета широкого круга задач для верификации приложения.



Спасибо за внимание!



Благодарности

- ♦ **Н.В. Никитину** за ценные замечания и конструктивную критику результатов работ
- ♦ **Вл.В. Воеводину** и **С.А. Жуматию** за предоставленный доступ к вычислительным ресурсам Суперкомпьютерного комплекса Московского университета и помощь в решении технических вопросов
- ♦ Компании **«Т-Сервисы»** за предоставленную возможность проведения ряда расчетов на вычислительной системе «Зилант»