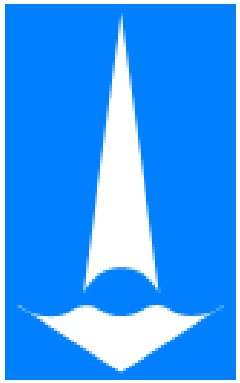
Применение графических ускорителей для расчета гидродинамических характеристик гребных винтов в пакете OpenFOAM



***Б.И. Краснопольский*[1,2]**

krasnopolsky@imec.msu.ru

***А.В. Медведев*[2]**

***А.Ю. Чулюнин*[1]**

[1]НИИ механики МГУ

[2]ЗАО “Т-Сервисы”

V международная конференция «Облачные вычисления. Образование. Исследования. Разработка»
Президиум РАН, Москва, 5 декабря 2014 г.

Пакет OpenFOAM

- ***Один из наиболее распространенных инженерных пакетов с открытым исходным кодом***
- ***Основное предназначение — решение задач гидрогазодинамики и теплообмена***

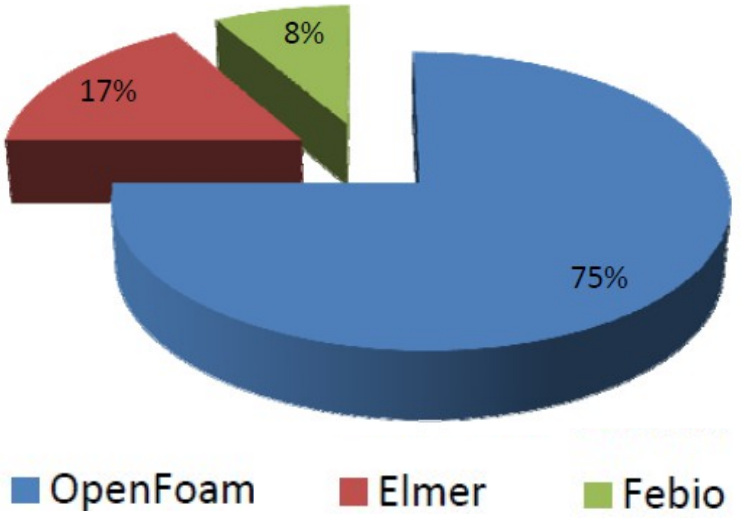
www.esi-group.com

  - Метод конечных объемов
  - Большой набор различных типов поддерживаемых сеток и численных схем
  - Обширный набор моделей турбулентности
  - Подвижные сетки
  - Для каждой математической модели создан отдельный решатель

- ***У пользователей есть возможность модификации уже имеющихся решателей и численных схем под свои нужды***

PRACE* initiative

- ***Опрос организаций промышленности, использующих инженерные пакеты с открытым исходным кодом***
  - Выделение наиболее широко используемых приложений
  - Предоставление доступа к вычислительным системам с установленным СПО
- ***Доработка СПО под перспективные вычислительные системы***
  - «…the main goal is to improve scalability of OpenFOAM for industrial relevant cases»
  - «As is typical of CFD applications the scalability bottleneck has been identified as being in the MPI communication pattern of the linear algebra core libraries.»



PRACE Second Implementation Project, RI-283493. D9.1.1. Support for Industrial Applications Year 1, 2012

**** Partnership for Advanced Computing in Europe***

Текущие достижения

- ***P. Dagna, J. Hertzer.*** **Evaluation of Multi-threaded OpenFOAM Hybridization for Massively Parallel Architectures.**

http://www.prace-project.eu/IMG/pdf/wp98.pdf

  - Создание гибридной реализации MPI+OpenMP для предобусловленного метода сопряженных градиентов и методов неполной факторизации
  - Гибридная модель оказалась **эффективнее** только для **1 из 24** расчетов

- ***M. Manguoglu.*** **A General Sparse Sparse Linear System Solver and Its Application in OpenFOAM.**

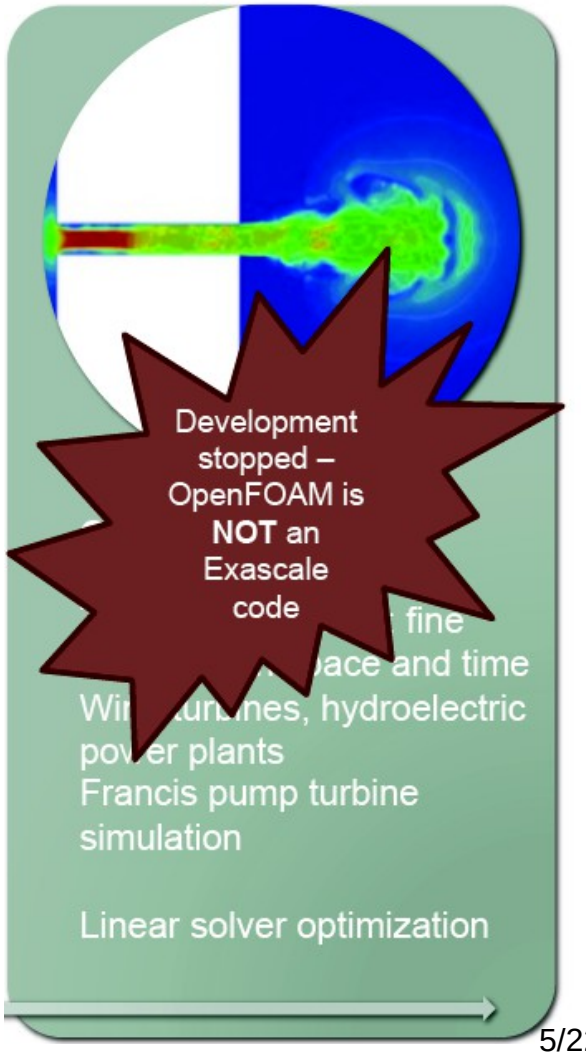
http://www.prace-ri.eu/IMG/pdf/A_General_Sparse_Sparse_Linear_System_Solver_and_Its_Application_in_OpenFOAM.pdf

  - Разработка новых численных методов для OpenFOAM, включая гибридную реализацию MPI+OpenMP
  - Неоднозначная эффективность по сравнению с классическими методами
  - Неудовлетворительные результаты масштабируемости

Вердикт

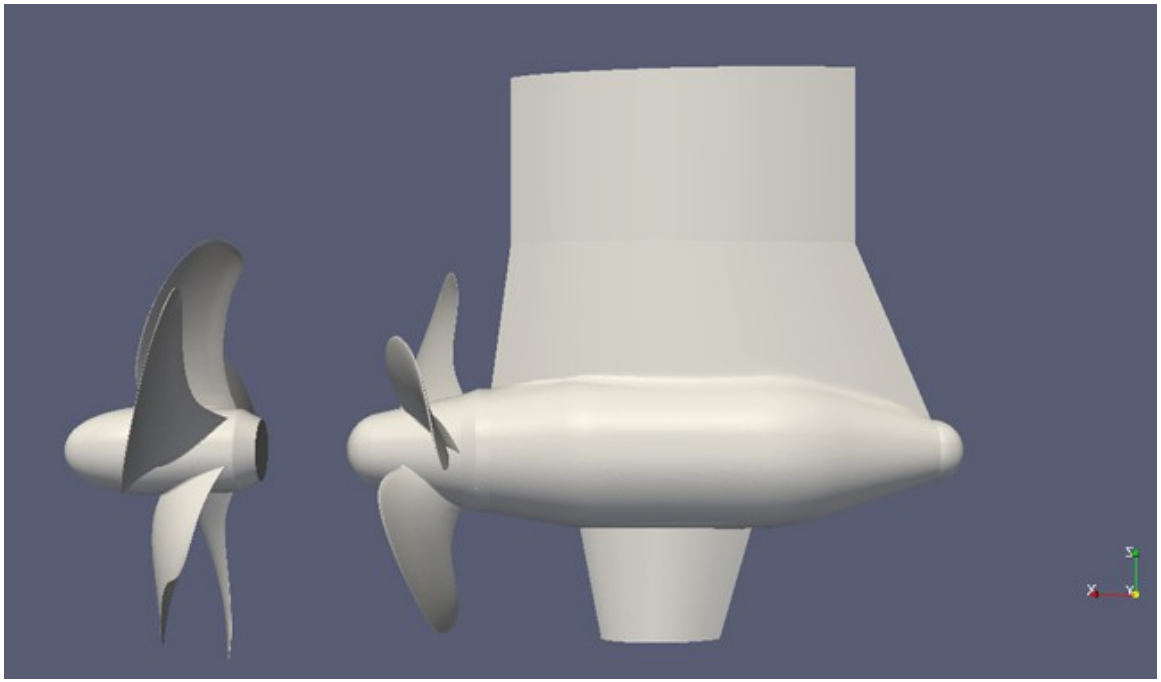
CRESTA — Collaborative Research into Exascale Systemware, Tools and Applications

- ***Доработка наиболее востребованных приложений для расчетов на перспективных вычислительных системах с миллионами вычислительных ядер (exascale)***
- ***M. Parsons.* Software co-design for extreme scale computing *// Extreme Scale Scientific Computing Workshop*, MSU, Moscow, Russia, 2014.**
  - OpenFOAM не пригоден для масштабных расчетов



Постановка целевой задачи

- ***Расчет гидродинамических характеристик пары гребных винтов судна***
  - Нестационарный расчет
  - Два вращающихся фрагмента сетки
  - k-ω SST модель турбулентности
  - Сетки порядка 100 млн. ячеек

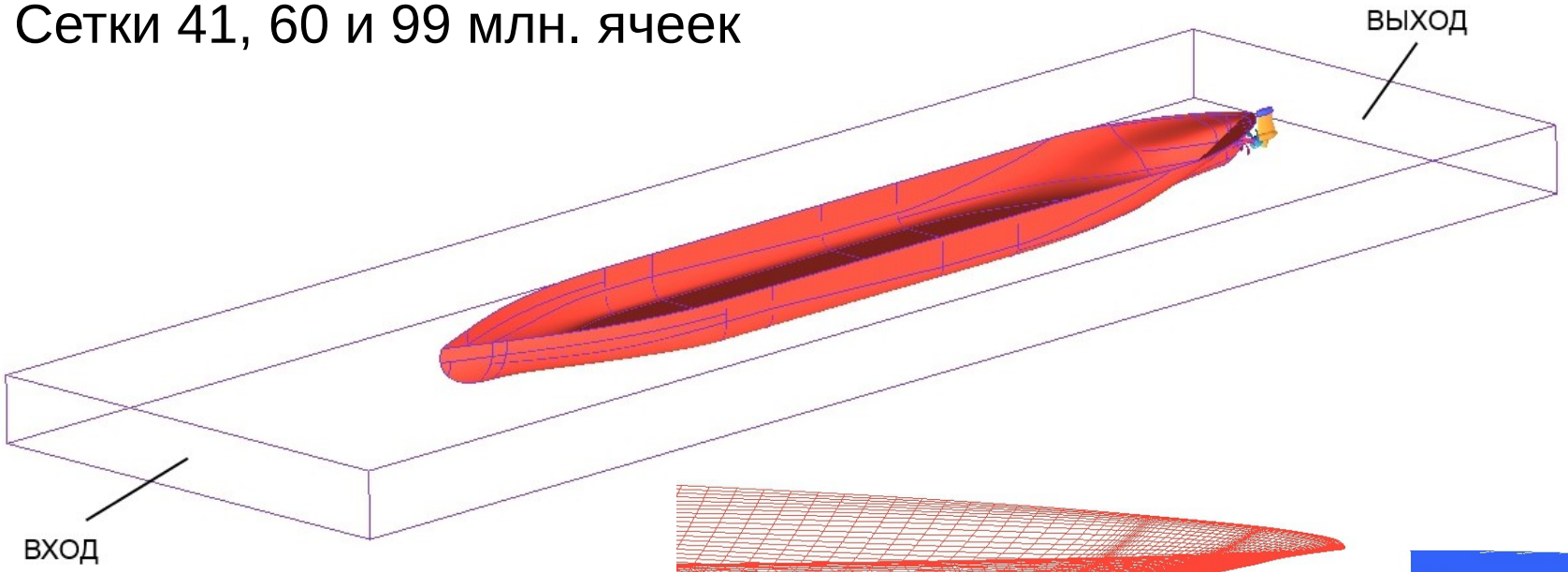


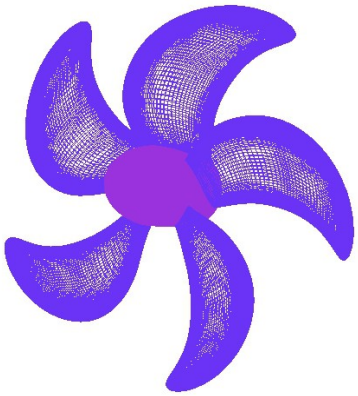
- ***Возможен ли расчет такой задачи в пакете OpenFOAM?***
- ***Какие могут быть предприняты дополнительные меры для ускорения таких расчетов в пакете OpenFOAM?***

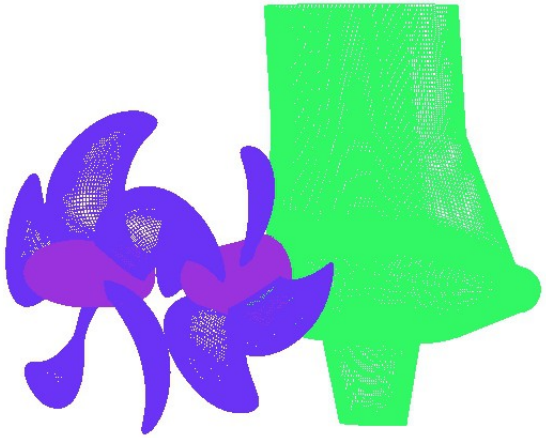
Постановка целевой задачи (2)

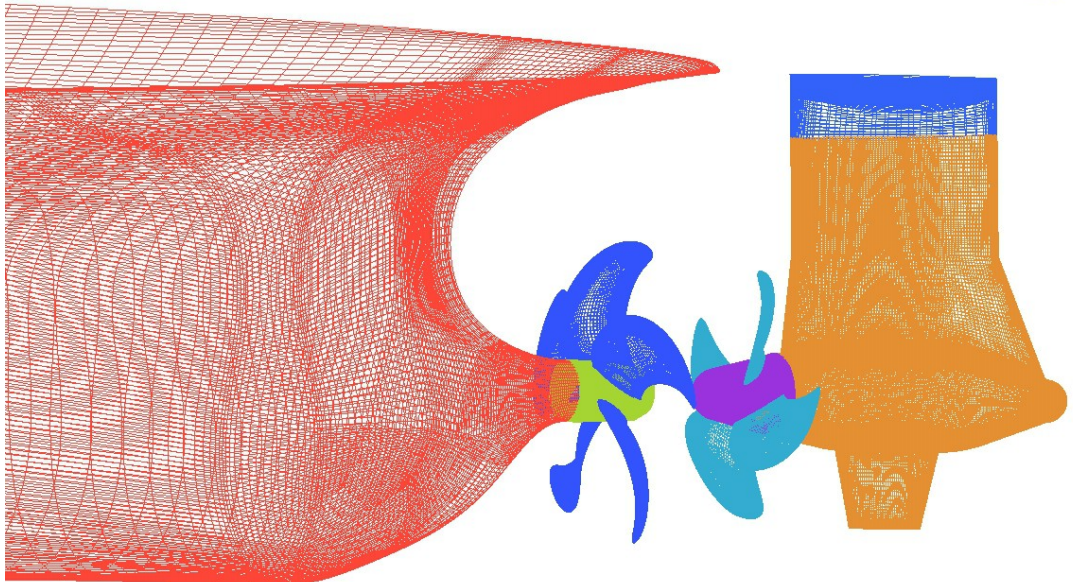
Расчет гидродинамических характеристик гребных винтов

Сетки 41, 60 и 99 млн. ячеек

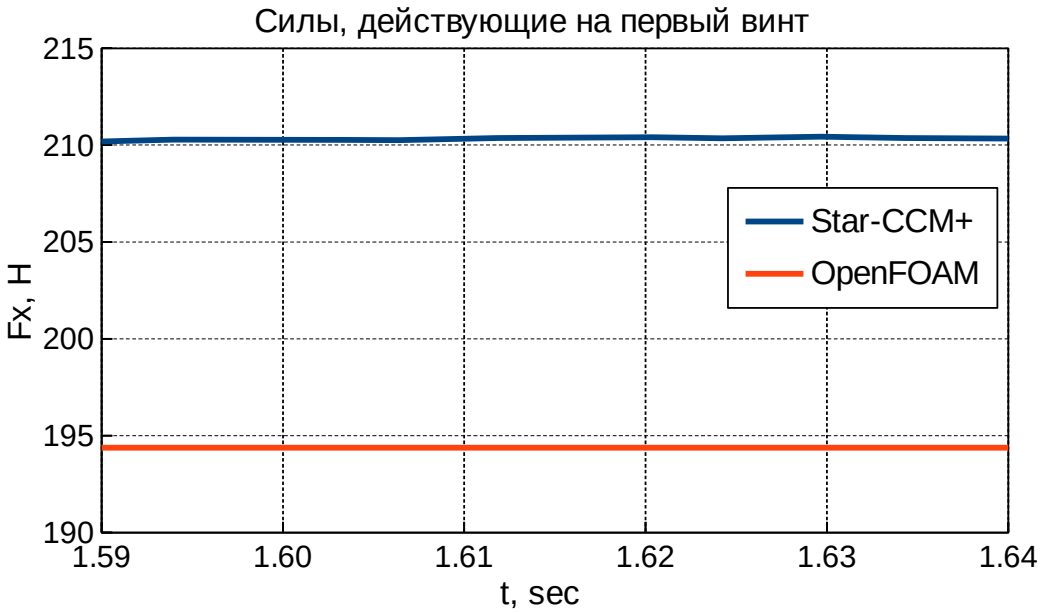


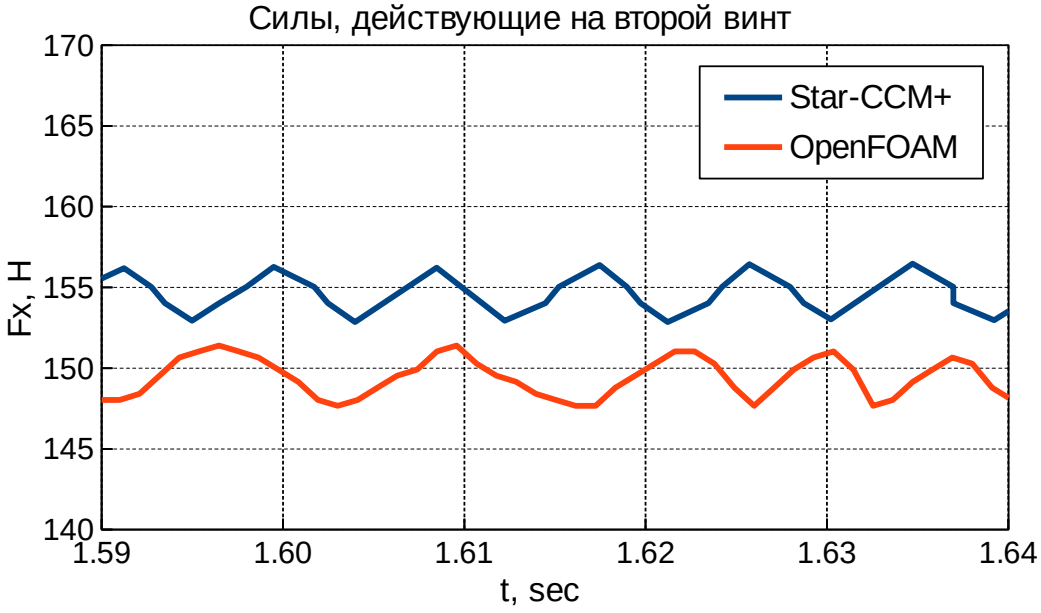


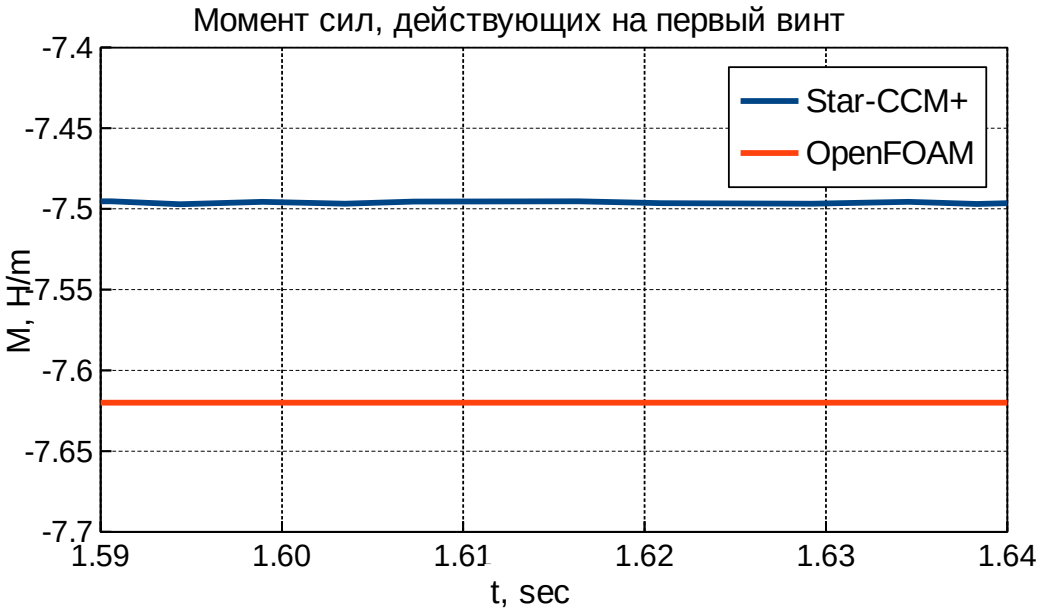




Сравнение с пакетом Star-CCM+







Стратегия ускорения расчетов

- ***Поскольку решение СЛАУ может занимать до 90% общего времени расчета задачи, можно сосредоточиться только на этапе решения СЛАУ***
- ***Решатель СЛАУ может быть реализован как плагин к OpenFOAM***
- ***Пути ускорения решения СЛАУ:***
  - Другие математические методы?
  - Гибридные модели программирования?
  - Использование сопроцессоров или ускорителей?

Решение СЛАУ для давления

- ***SparseLinSol*** - библиотека для решения больших разреженных СЛАУ
- Плагин, позволяющий использовать реализованные методы при расчете задач в пакете OpenFOAM, в т.ч. в MultiGPU-режиме
  - Поддержка ***processor***- и ***cyclicAMI***-патчей для расчета задач с вращающимися подобластями в параллельном режиме

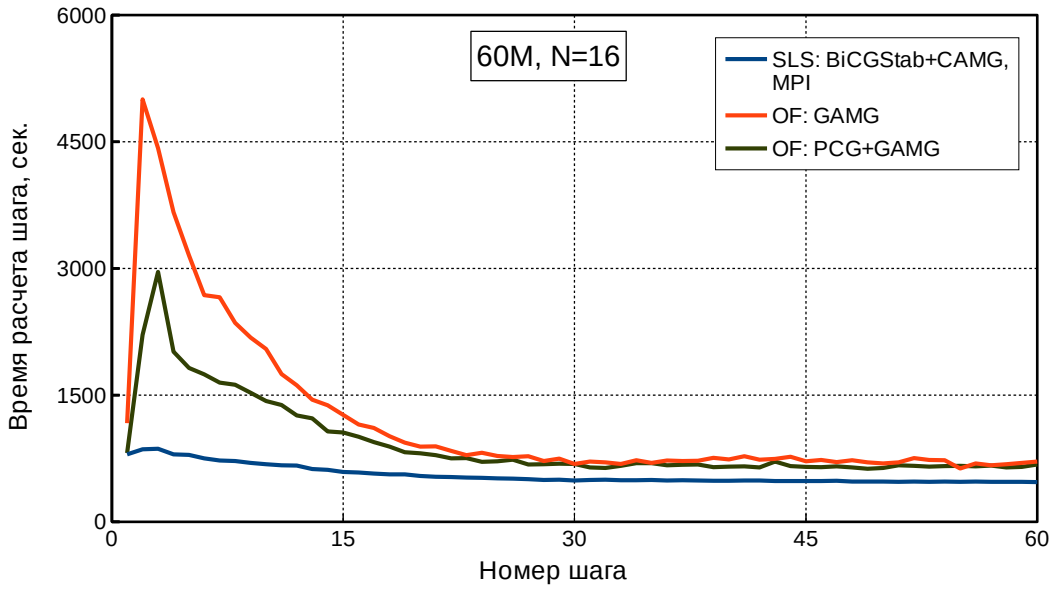
Методы из пакета OpenFOAM

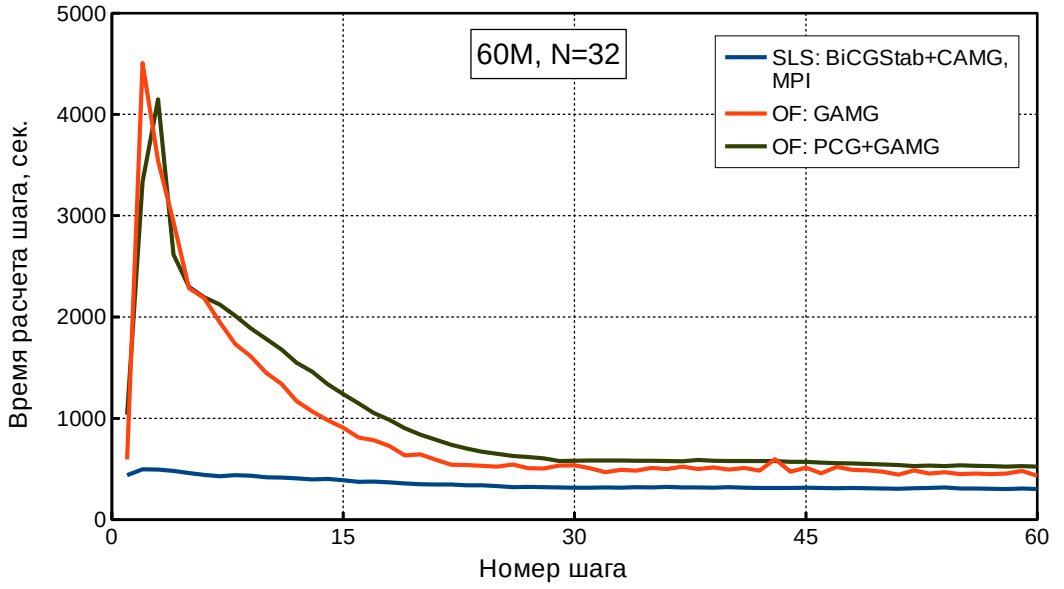
- **GAMG:** геометрический-алгебраический многосеточный метод
- **PCG+GAMG:** метод сопряженных градиентов с геометрическим-алгебраическим многосеточным методом в качестве предобуславливателя
  - сглаживатель: метод Холецкого

Реализованные методы

- **BiCGStab+CAMG:** стабилизированный метод бисопряженных градиентов с классическим алгебраическим многосеточным предобуславливателем
- **Два набора параметров:**
  - (1) сглаживатель: симметричный метод Гаусса-Зейделя
  - (2) сглаживатель: метод итераций Чебышева

Время расчёта одного шага по времени





СК “Ломоносов”

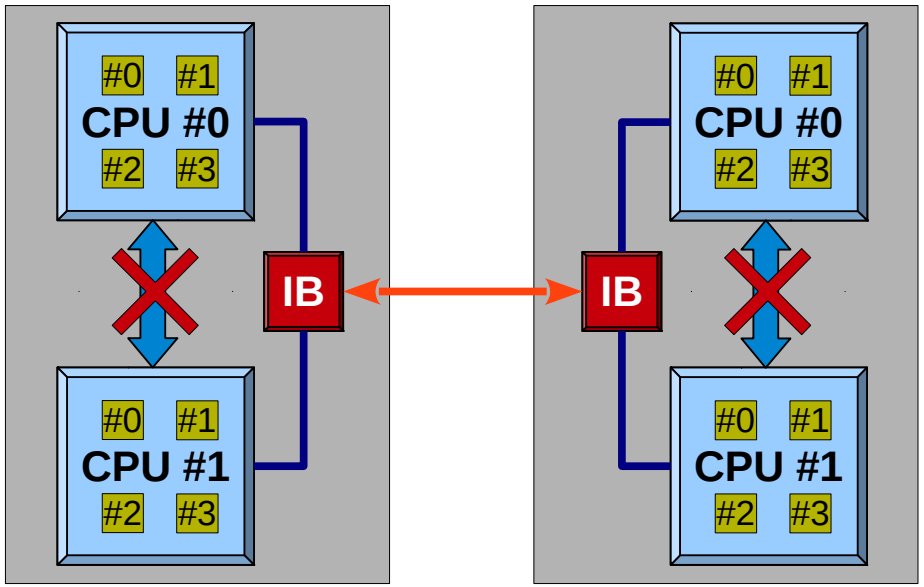
SLS: BiCGStab + CAMG, GS	OF: GAMG	OF: PCG + GAMG
Суммарное время расчета, час:		
9.2	19.8	15.3
Характерное время расчета шага, сек.		
480	700	680
Суммарное время расчета, час:		
5.9	14.4	16.4
Характерное время расчета шага, сек.		
310	460	530

Гибридные модели

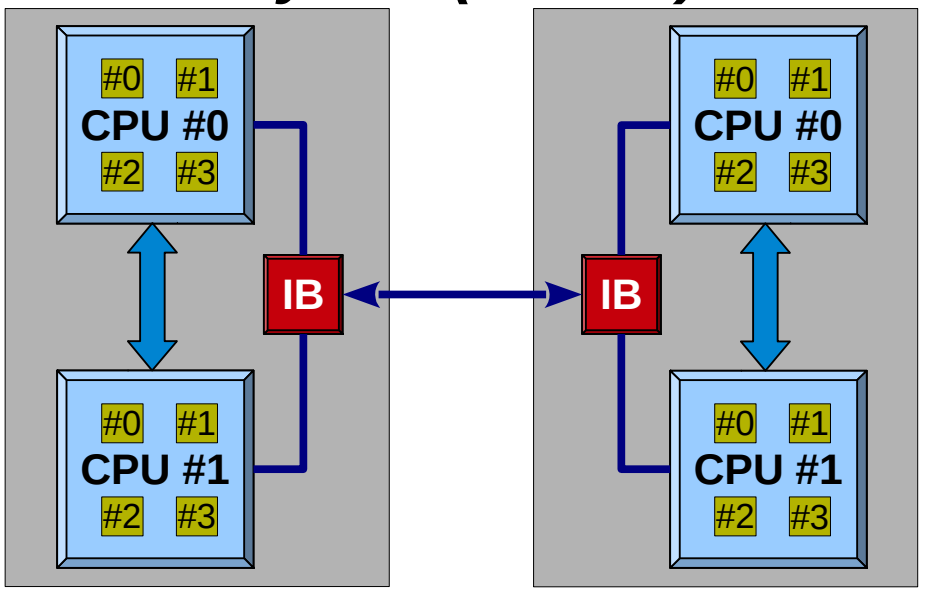
Типичный вычислительный узел: два процессора, каждый процессор содержит от 4 до 16 вычислительных ядер

Пример: 2 вычислительных узла по 8 ядер, обмен сообщениями между вычислительными процессами («каждый с каждым»)

MPI



Hybrid (MPI+...)



Количество сообщений, пересылаемых по коммуникационной сети:

$N_{msg} = NP * (NP-NC) = 16 * (16-8) = 128$

$N_{msg} = (NP/NC) * (NP/NC-1) = 2 * (2-1) = 2$

Гибридные модели (2)

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

- ***Необходимо адаптировать вычислительные методы и разрабатывать многоуровневые алгоритмы распараллеливания:***
  - Двухуровневые: **node / core**
  - Трехуровневые: **node / numa-node / core**
  - Четырехуровневые: **node / CPU-socket / numa-node / core**

GPU device

«T-Nano», MPI vs MPI+ShM

- *Сравнение двух реализаций методов*
- *Время, затраченное в ходе расчета одного шага на решение СЛАУ для давления, сек. (3-й шаг)*

Кол-во ядер	SLS: MPI			SLS: MPI+ShM		
	SLS: Setup	SLS: Solve	SLS: Total	SLS: Setup	SLS: Solve	SLS: Total
128	370	1032	1402	440	872	1312
256	170	517	697	220	411	631
384	135	442	577	165	298	463
Ускорение:	2.74	2.33	2.43	2.67	2.93	2.83

Линейная масштабируемость solve-части методов для гибридной модели

GPU как ускоритель

Особенности реализации методов на GPU:

- ***Перенести код целиком невозможно***
  - Выделение ключевых блоков и реализация в виде ядер CUDA
- ***Другая архитектура памяти – пересмотр всех алгоритмов***
- ***Копирование данных между памятью узла и ускорителем***
  - Дополнительная латентность вызовов
  - Влияет на организацию процедуры обмена сообщениями через MPI
- ***Задача относится к категории memory-bound: время выполнения операции определяется пропускной способностью памяти***
- ***Макс. пропускная способность памяти устройств (СК «Нижеголь»)***
  - Intel Xeon E5-2665, 1 socket: *51.2 Gb/s*
  - NVIDIA Tesla M2090, ECC off: 177 Gb/s
    ECC on: *155 Gb/s*

Соотношение: 155 / 51.2 ≈ 3.0 теоретический максимум ускорения за счет GPU на данной системе

Оптимизации для GPU

Особенности оптимизации методов на GPU:

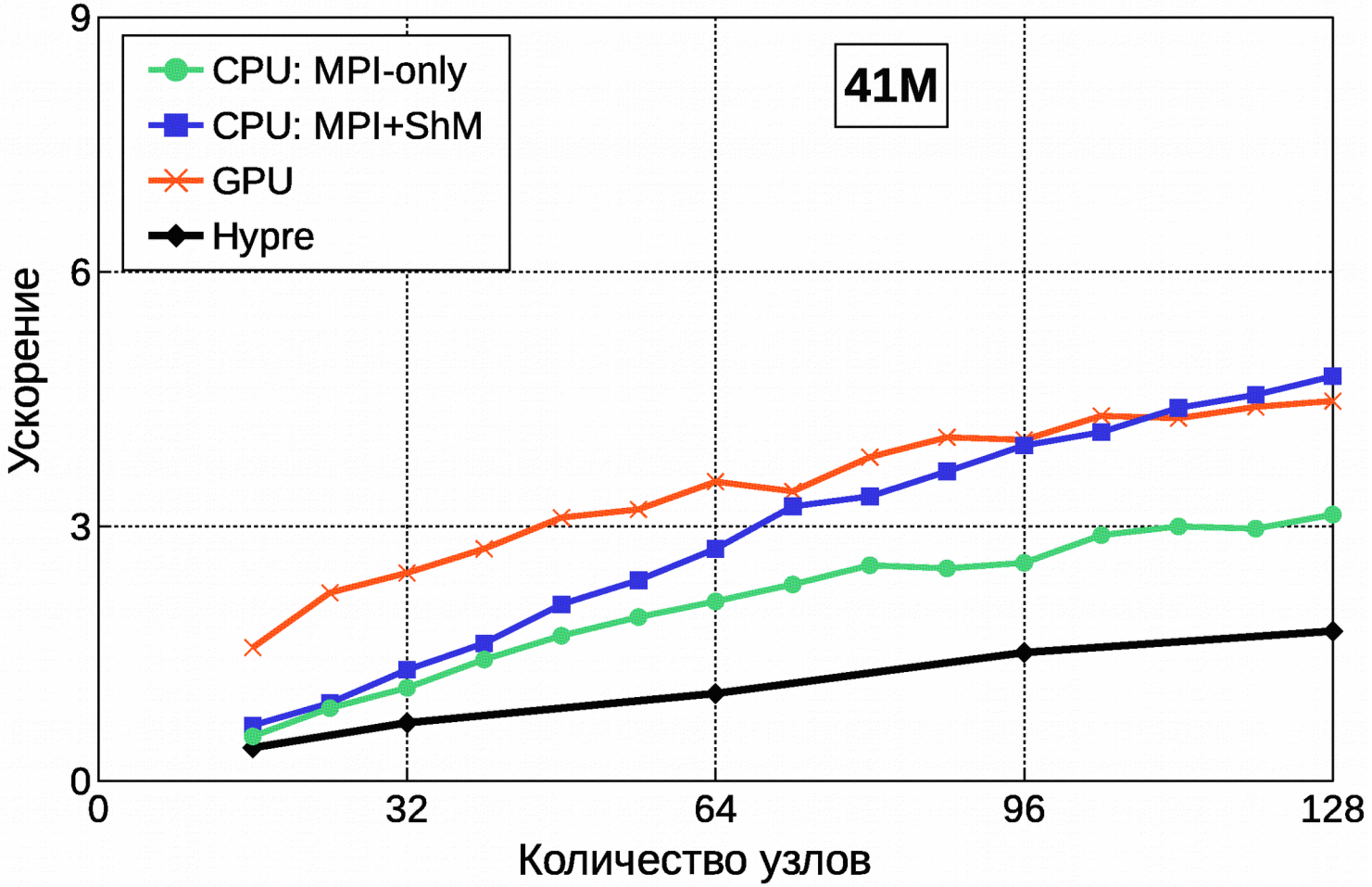
- ***Форматы представления матриц***
  - **CSR**: наиболее универсальный формат, экономично представляет любой тип разреженной матрицы
  - **ELLPACK**: более экономичный формат, оптимален для матриц с одинаковым числом ненулевых элементов в строках матрицы
  - **DIA**: наиболее экономичный формат, оптимален для диагональных матриц
  - Гибридные форматы: матрица представлена в виде комбинаций форматов
- ***Эффективность на GPU:***
  - **CSR** в целом не эффективен, нет хорошего алгоритма для GPU
  - **ELLPACK** достаточно эффективен для подходящих типов матриц
  - Хорошие результаты даёт только комбинирование форматов, например **CSR+ELLPACK**

Оптимизации для GPU (2)

Особенности оптимизации методов на GPU:

- ***Смешанная точность (64 bit+32 bit) представления матриц***
  - Актуально как на CPU (ускорение 10-15%), так и на GPU (ускорение 15-20%)
- ***Асинхронность операций копирования CPU<->GPU, вычислений и MPI-обменов***
  - Требует очень аккуратного программирования и нестандартных решений
- ***Часть уровней многосеточного метода целесообразно оставлять на CPU***
  - Для матриц малого размера GPU теряет эффективность настолько, что латентность операций с GPU превышает выгоды от его использования
  - Выигрыш варьируется от 10% до 30%
- ***Архитектура Kepler отличается от Fermi, требуется адаптация***

SparseLinSol, масштабируемость, 41M



СК **“Ломоносов”**:

CPU: Intel Xeon E5630 8 cores/node

GPU: NVidia Tesla X2070, 2 GPUs/node

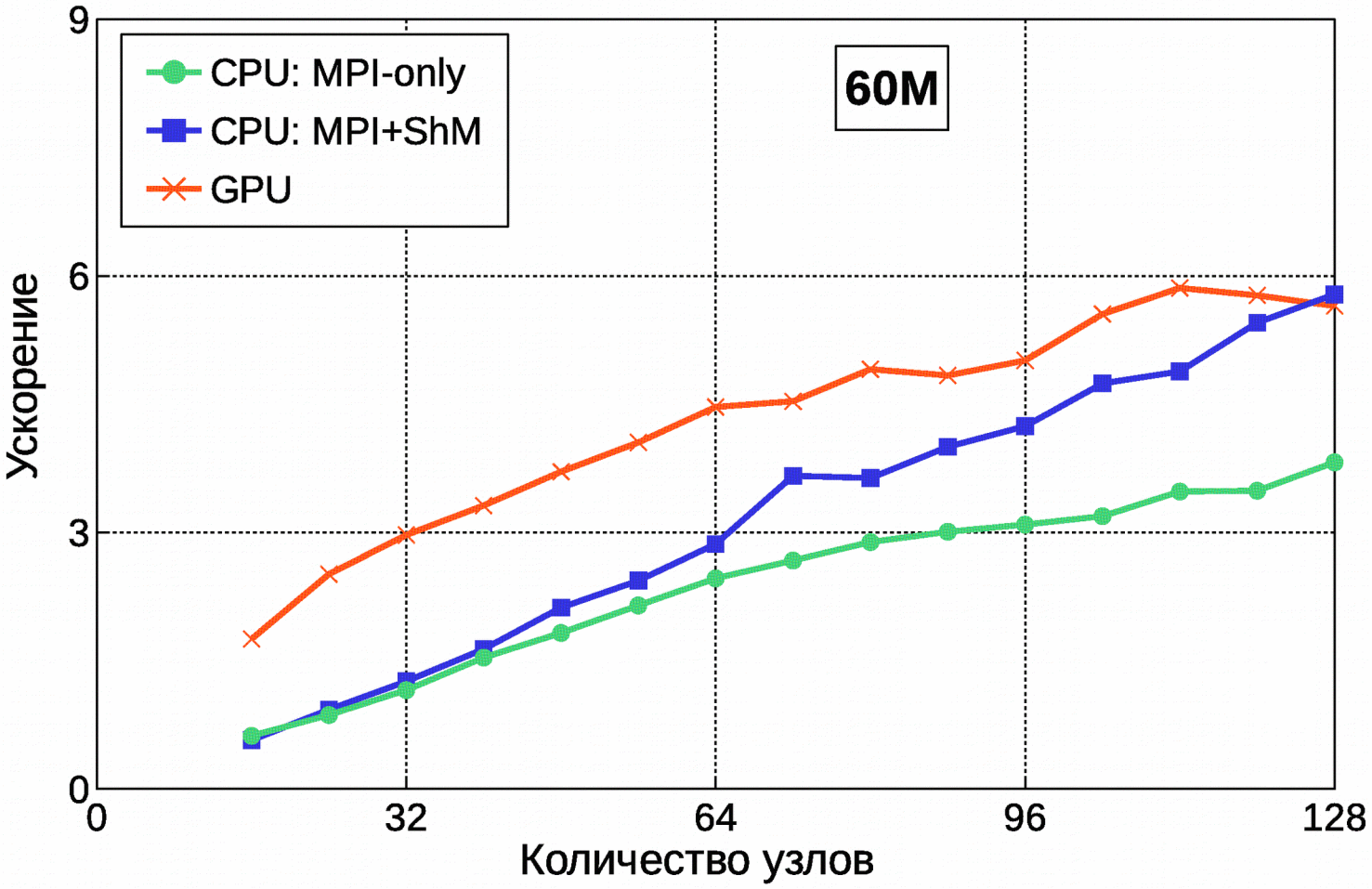
- **Кратное преимущество для MPI и MPI+ShM относительно hypre**
- **2-кратный выигрыш за счет GPU на малом количестве узлов**

Матрица: 41 млн. неизвестных (гребные винты)

Метод: BiCGStab + CAMG, сглаж. Чебышёва

Результаты нормированы на время расчета на 32 узлах, *“CPU: MPI-only”*

SparseLinSol, масштабируемость, 60M



СК **“Ломоносов”**:

CPU: Intel Xeon E5630 8 cores/node

GPU: NVidia Tesla X2070, 2 GPUs/node

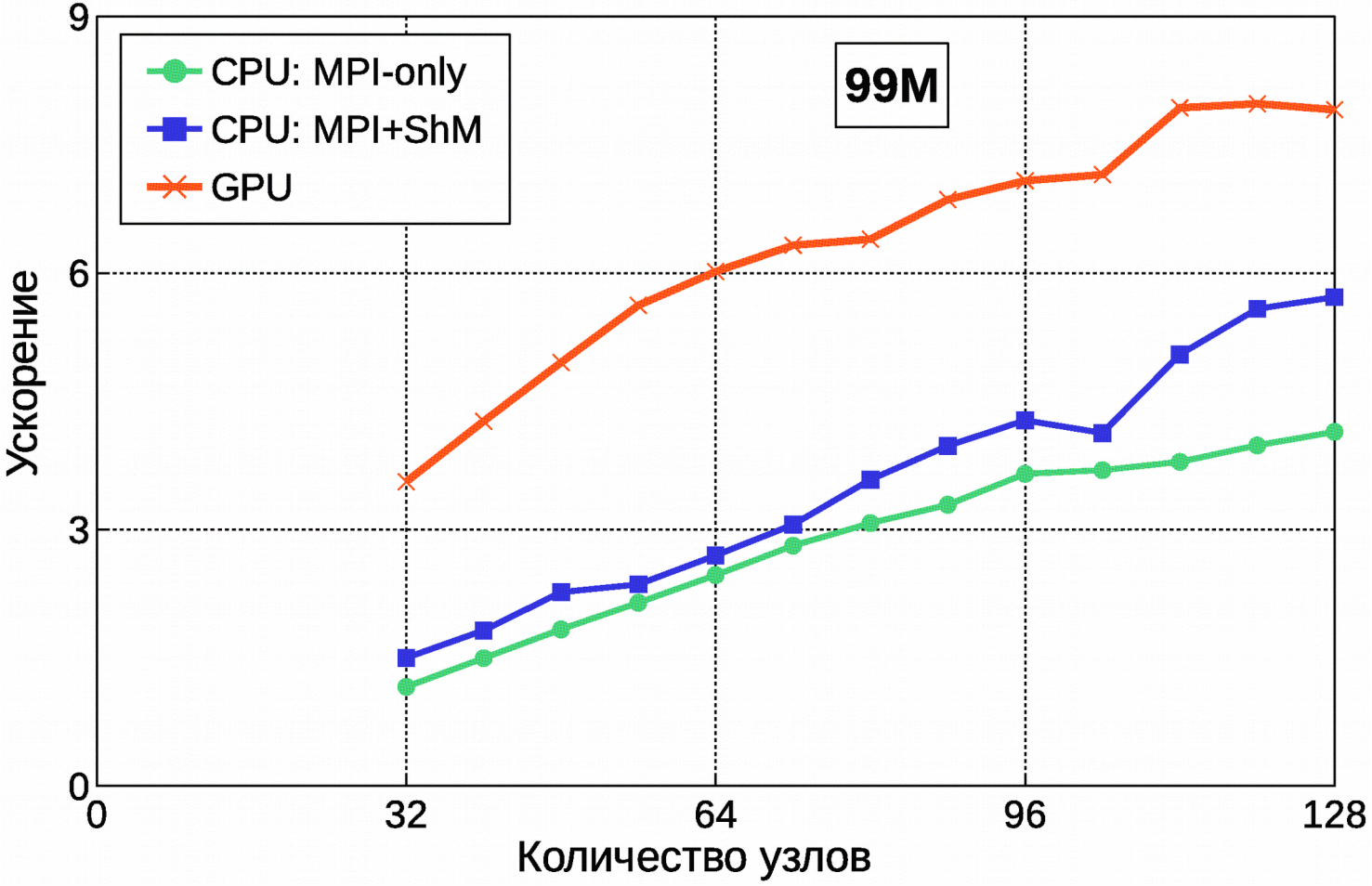
- **Ускорение до 1.7 раза за счет гибридной модели**
- **GPU обеспечивает ускорение во всем диапазоне узлов**

Матрица: 60 млн. неизвестных (гребные винты)

Метод: BiCGStab + CAMG, сглаж. Чебышёва

Результаты нормированы на время расчета на 32 узлах, *“CPU: MPI-only”*

SparseLinSol, масштабируемость, 99M



СК **“Ломоносов”**:

CPU: Intel Xeon E5630 8 cores/node

GPU: NVidia Tesla X2070, 2 GPUs/node

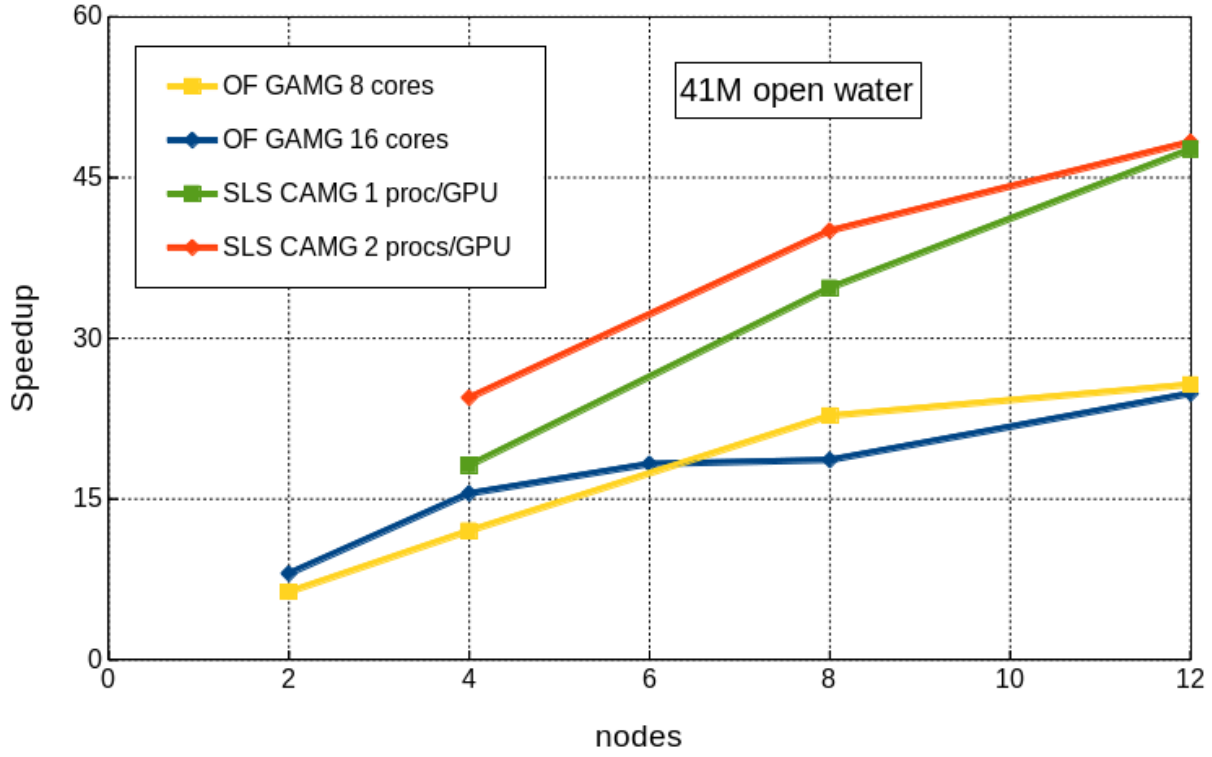
- **Линейная масштабируемость гибридной модели**
- **1.4-3 кратное ускорение за счет использования GPU**

Матрица: 99 млн. неизвестных (гребные винты)

Метод: BiCGStab + CAMG, сглаж. Чебышёва

Результаты нормированы на время расчета на 32 узлах, *“CPU: MPI-only”*

Использование плагина для OpenFOAM



СК **“Нижеголь”**:

CPU: Intel Xeon E5-2665 16 cores/node

GPU: NVidia Tesla M2090, 2 GPUs/node

- ***Реализованные методы более робастны по сравнению с методами из пакета OpenFOAM***
- ***Удаётся ускорить реальные расчёты больших гидродинамических задач в OpenFOAM в два раза и более***
- ***Обширные перспективы дальнейшего ускорения...***

Спасибо за внимание!