

**USING AN INTEGRATED APPROACH FOR
SOLVING LARGE SIZE PHYSICS
PROBLEMS ON SUPERCOMPUTERS**



B.M. Glinskiy, I.M. Kulikov, I.G. Chernykh, A.V. Snytnikov, A.F. Sapetina, D.V. Vins

СОДЕРЖАНИЕ

- АКТУАЛЬНОСТЬ РАБОТЫ
- ПРЕДЛАГАЕМЫЙ ПОДХОД
- ПРИМЕРЫ: СЕЙСМИКА, АСТРОФИЗИКА, ФИЗИКА ПЛАЗМЫ
 - ✓ СО-ДИЗАЙН ПАРАЛЛЕЛЬНЫХ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ
 - ✓ ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ
 - ✓ ЭНЕРГОЭФФЕКТИВНОСТЬ АЛГОРИТМОВ
- ЗАКЛЮЧЕНИЕ

АКТУАЛЬНОСТЬ РАБОТЫ

- Сложность применения будущих экзафлопсных суперкомпьютеров будет связана с невероятной степенью параллелизма, это сотни миллионов и миллиарды одновременно работающих вычислительных ядер.
- Уровень развития супервычислений для машин этого класса связан как с разработкой структур самих суперкомпьютеров, так и с совершенствованием их алгоритмического и программного обеспечения.
- Вычислительные алгоритмы, как правило, являются более консервативными по сравнению с развитием средств вычислительной техники.
- Пока не будет разработана специальная математическая основа, учитывающая особенности экзафлопсных суперкомпьютеров будут применяться алгоритмы, реализованные на современных суперкомпьютерах.
- Важно уже сейчас, на стадии разработки экзафлопсных суперкомпьютеров, оценить перспективы их реализации, исследовать параллелизм, масштабируемость и энергопотребление.

ПРЕДЛАГАЕМЫЙ ПОДХОД

Предлагается методика разработки алгоритмического и программного обеспечения для суперкомпьютеров, содержащий три связанных этапа:

- 1) со-дизайн, под которым мы понимаем адаптацию постановки задачи, математического метода, вычислительного алгоритма под параллельную архитектуру суперкомпьютера на всех этапах решения задачи;
- 2) создание упреждающего алгоритмического и программного обеспечения для наиболее перспективных суперкомпьютеров на основе имитационного моделирования;
- 3) оценка энергоэффективности алгоритмов при различных реализациях на заданной архитектуре суперЭВМ.

Глинский Б.М., Куликов И.М., Снытников А.В., Черных И.Г., Винс Д.В. Многоуровневый подход к разработке алгоритмического и программного обеспечения экзафлопсных суперЭВМ // Вычислительные методы и программирование. – 2015. – Т. 16. – С. 543-556.

КО-ДИЗАЙН МЕТОДОВ РЕШЕНИЯ ВЫЧИСЛИТЕЛЬНО-СЛОЖНЫХ ЗАДАЧ

Понятие ко-дизайна в зарубежной литературе часто применяется при разработке архитектуры компьютеров (*Ang J. et al. High Performance Computing: From Grids and Clouds to Exascale, chapter Exascale Computing and the Role of Co-design. IOS Press Inc, 2011*).

Этот путь является заведомо тупиковым, так как не учитывает развитие программного обеспечения для подобных архитектур (*Barret R. et al. On the Role of Co-design in High Performance Computing. Transition of HPC Towards Exascale Computing. IOS Press, 2013*).

В последнее время под ко-дизайном понимается совместная разработка архитектуры и программного обеспечения (Hardware/Software co-design).

Для реализации такой концепции созданы специальные рабочие группы в лаборатории Лос-Аламоса (<http://codesign.lanl.gov/>) и Ливерморской лаборатории (<https://codesign.llnl.gov/>).

С 2011 года проводятся международные конференции по различным сторонам концепции ко-дизайна (<http://www.ncic.ac.cn/codesign/>).

СО-ДИЗАЙН МЕТОДОВ РЕШЕНИЯ ВЫЧИСЛИТЕЛЬНО-СЛОЖНЫХ ЗАДАЧ

Авторы доклада предлагают расширить этот термин применительно к разработке алгоритмического и программного обеспечения суперкомпьютеров. В данном случае **основой понятия со-дизайна является анализ эффективности параллельной реализации на всех этапах решения задачи под заданную архитектуру суперкомпьютера:**

- 1) формулировка «физической» постановки задачи;
- 2) математическая формулировка «физической» задачи;
- 3) создание численного метода для решения математической постановки;
- 4) выбор структур данных и параллельного алгоритма;
- 5) учёт особенностей архитектуры суперЭВМ;
- 6) использование средств разработки.

*Boris Glinskiy, Igor Kulikov, Alexey Snytnikov, Alexey Romanenko, Igor Chernykh, Vitaly Vshivkov.
Co-design of Parallel Numerical Methods for Plasma Physics and Astrophysics.*

Supercomputing frontiers and innovations. Vol 1, No 3 (2014), pp. 88-98.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ АЛГОРИТМОВ

Исследование масштабируемости параллельных алгоритмов проводится с применением мультиагентного распределённого моделирования систем с дискретными событиями **AGNES** (AGent based NEtwork Simultor), разработанной в ИВМиМГ СО РАН.

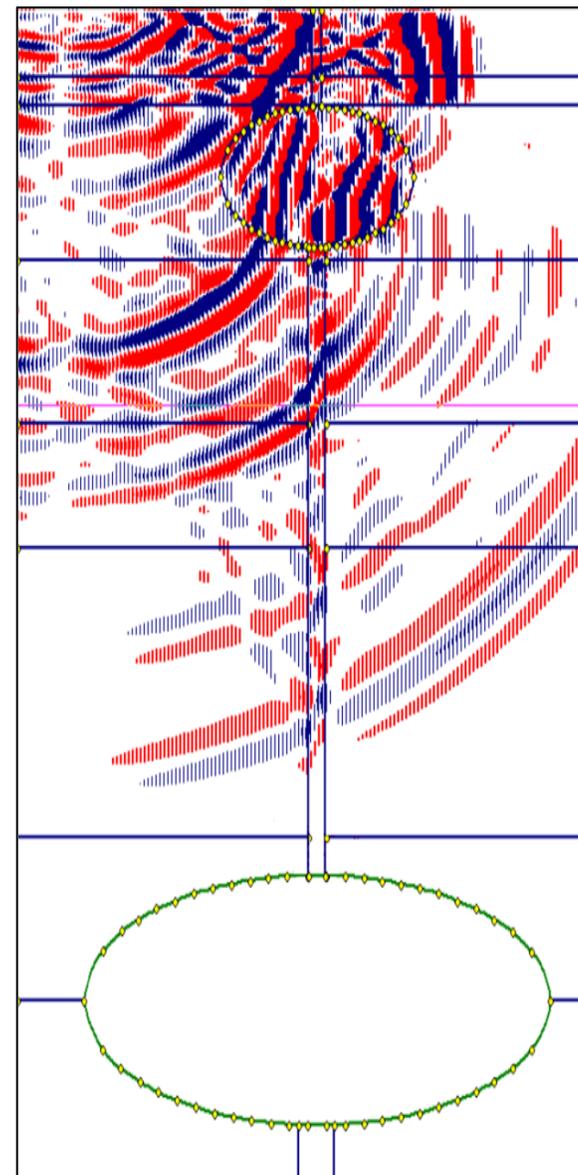
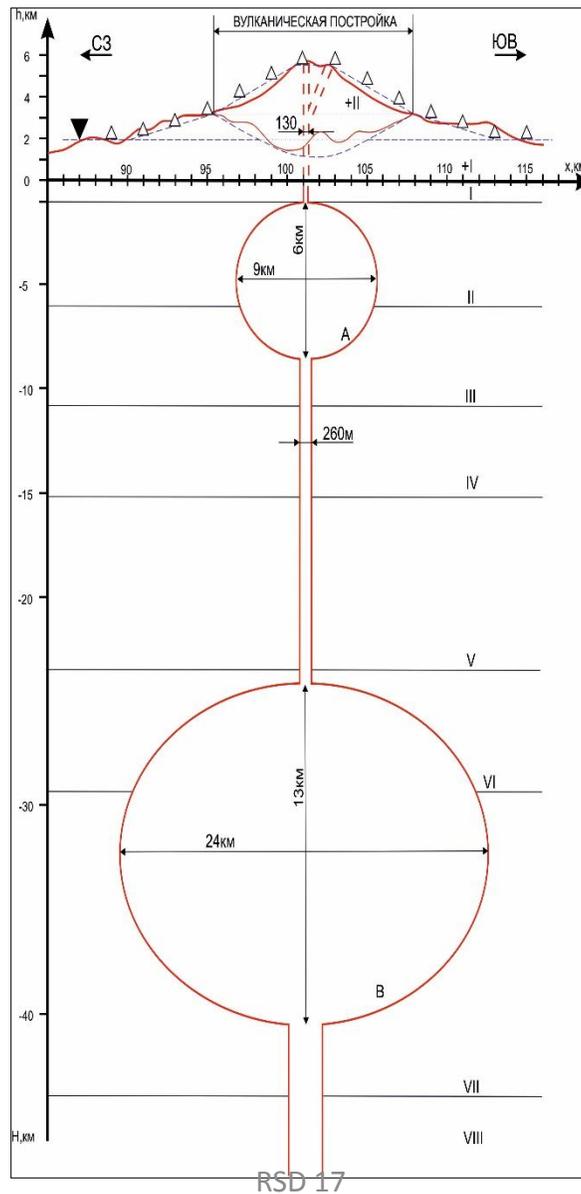
Основу системы представляют три группы агентов: системные, отвечающие за функционирование модели в конкретной операционно-вычислительной среде; управляющие, отвечающие за отказоустойчивое имитационное моделирование, включая управление синхронизацией событий и сбора статистики; проблемно-ориентированные, имитирующие работы компонентов моделируемых программ и суперЭВМ.

Б.М. Глинский, М.А. Марченко, Б. Г. Михайленко, А.С. Родионов, И.Г. Черных, Д.А. Караваев, Д.И. Подкорытов, Д. В. Винс. Отображения параллельных алгоритмов для суперкомпьютеров экзафлопсной производительности на основе имитационного моделирования // Информационные технологии и вычислительные системы, 2013, № 4, с. 3-14.

ЭНЕРГОЭФФЕКТИВНОСТЬ АЛГОРИТМОВ И ПРОГРАММ

- Параллельно развитию энергоэффективности аппаратных платформ, актуально исследование энергоэффективности алгоритмов и программ, которые сводятся к вычислительной эффективности, эффективности обработки данных, информированности о контексте вычислительной среды.
- С целью повышения энергоэффективности алгоритмов для задач сейсмологии, астрофизики и физики плазмы каждый программный код оптимизировался по следующим трем направлениям: эффективность работы с процессором и оперативной памятью, эффективность работы с сетевыми устройствами и эффективность работы с периферическими системами.
- Для оптимизации эффективной работы с процессором и памятью использовался пакет Intel Parallel Studio XE 2016 Beta, включающий в себя Intel Vectorization Advisor, который анализирует код и выдает предложения для оптимизации программы с целью максимизации эффективности использования векторных инструкций. Использование Intel Vectorization Advisor позволило ускорить астрофизический код на 30%.
- Для задач сейсмологии, астрофизики и физики плазмы были выбраны наиболее эффективные численные методы с точки зрения минимизации сетевых обменов.

МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ СЕЙСМИЧЕСКИХ ВОЛН В СРЕДАХ, ХАРАКТЕРНЫХ ДЛЯ ВУЛКАНИЧЕСКИХ СТРУКТУР



ПОСТАНОВКА ЗАДАЧИ В ТЕРМИНАХ СКОРОСТЕЙ СМЕЩЕНИЯ И НАПРЯЖЕНИЯ

$$\rho \frac{\partial \vec{u}}{\partial t} = [A] \vec{\sigma} + \vec{F}(t, x, y, z), \quad \frac{\partial \vec{\sigma}}{\partial t} = [B] \vec{u},$$

$$A = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial z} \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}, \quad B = \begin{bmatrix} (\lambda + 2\mu) \frac{\partial}{\partial x} & \lambda \frac{\partial}{\partial y} & \lambda \frac{\partial}{\partial z} \\ \lambda \frac{\partial}{\partial x} & (\lambda + 2\mu) \frac{\partial}{\partial y} & \lambda \frac{\partial}{\partial z} \\ \lambda \frac{\partial}{\partial x} & \lambda \frac{\partial}{\partial y} & (\lambda + 2\mu) \frac{\partial}{\partial z} \\ \mu \frac{\partial}{\partial y} & \mu \frac{\partial}{\partial x} & 0 \\ \mu \frac{\partial}{\partial z} & 0 & \mu \frac{\partial}{\partial x} \\ 0 & \mu \frac{\partial}{\partial z} & \mu \frac{\partial}{\partial y} \end{bmatrix},$$

$$\sigma|_{t=0} = 0, \quad u|_{t=0} = 0, \quad \sigma_{xz}|_{z=0} = 0, \quad \sigma_{yz}|_{z=0} = 0, \quad \sigma_{zz}|_{z=0} = 0$$

$\vec{u} = (u, v, w)^T$ – вектор скоростей смещений,

$\vec{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T$ – тензор напряжений,

$\rho(x, y, z)$ – плотность, $\lambda(x, y, z), \mu(x, y, z)$ – параметры Ламе,

$\vec{F}(t, x, y, z)$ – массовая сила

ПОСТАНОВКА ЗАДАЧИ В ТЕРМИНАХ СМЕЩЕНИЙ

$$\rho \frac{\partial^2 \vec{U}}{\partial t^2} = [C] \vec{U} + \vec{F}(t, x, y, z)$$

$$C = \begin{bmatrix} (\lambda + 2\mu) \frac{\partial^2}{\partial x^2} + \mu \left(\frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) & (\lambda + \mu) \frac{\partial^2}{\partial x \partial y} & (\lambda + \mu) \frac{\partial^2}{\partial x \partial z} \\ (\lambda + \mu) \frac{\partial^2}{\partial y \partial x} & (\lambda + 2\mu) \frac{\partial^2}{\partial y^2} + \mu \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \right) & (\lambda + \mu) \frac{\partial^2}{\partial y \partial z} \\ (\lambda + \mu) \frac{\partial^2}{\partial z \partial x} & (\lambda + \mu) \frac{\partial^2}{\partial z \partial y} & (\lambda + 2\mu) \frac{\partial^2}{\partial z^2} + \mu \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \end{bmatrix}$$

$$U(x, y, z) |_{t=0} = 0, \quad \sigma_{xz} |_{z=0} = 0, \quad \sigma_{yz} |_{z=0} = 0, \quad \sigma_{zz} |_{z=0} = 0$$

$\vec{U} = (U, V, W)^T$ – тензор напряжений,

$\rho(x, y, z)$ – плотность, $\lambda(x, y, z), \mu(x, y, z)$ – параметры Ламе,

$\vec{F}(t, x, y, z)$ – массовая сила

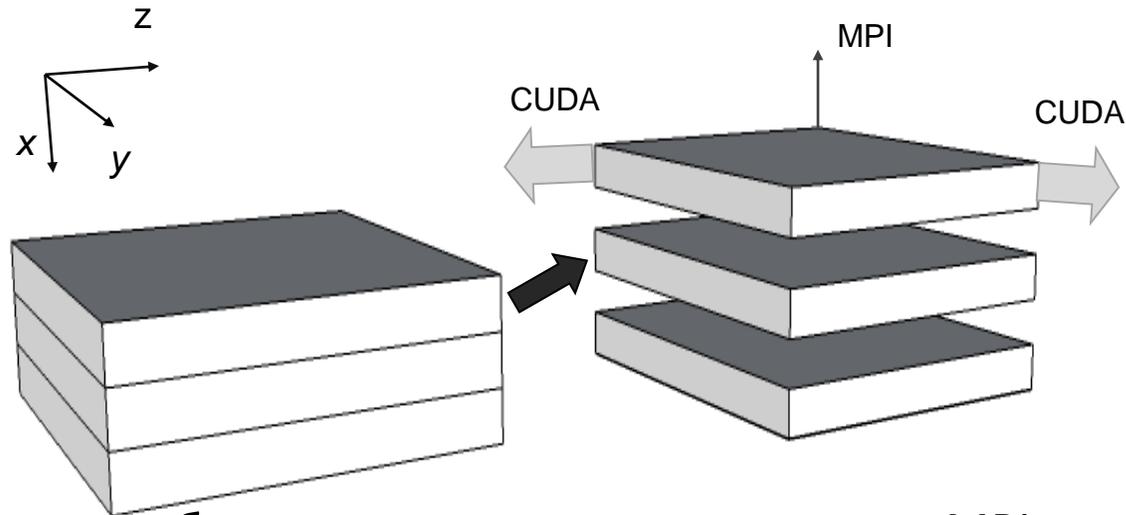
МЕТОД РЕШЕНИЯ

- Конечно-разностный метод
- Явные конечно-разностные схемы на сдвинутых сетках*
- Второй порядок аппроксимации по времени и пространству
- Равномерные сетки
- Критерий устойчивости:
$$\tau \leq \left(Vp_{\max} \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} \right)^{-1},$$

где Vp_{\max} – максимальная скорость распространения упругих волн

* Подход к построению схем описан в статье: Bihn M., Weiland T. A Stable Discretization Scheme for the Simulation of Elastic Waves // Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics (IMACS 1997). Т. 2, С. 75-80.

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ: ДЕКОМПОЗИЦИЯ И ОПТИМИЗАЦИЯ РАБОТЫ С ПАМЯТЬЮ



- Обмены между узлами: неблокированная передача данных MPI
- Обмены между графическими картами: асинхронные функции CUDA
- Оптимальное расположение всех используемых трехмерных массивов и соответствующее распределение работ между нитями.
- Сохранение в константную память графической карты всех основных констант, используемых на каждом шаге по времени.
- В памяти GPU хранятся не исходные сеточные коэффициенты, а их модификации. Реализация для **расчета скоростей смещения и напряжения** требует выделения памяти для хранения **26** трехмерных массивов, а реализация для **расчета смещений всего 14 массивов**.

СРАВНЕНИЕ ВРЕМЕНИ РАСЧЕТОВ

	$\vec{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T$ $\vec{u} = (u, v, w)^T$	$\vec{U} = (U, V, W)^T$
Кол-во уравнений	9	3
Кол-во 3D массивов в программе	26	14
Кол-во операций с плавающей точкой на один временной шаг	57	98
Время расчета на 15 узлах кластера НКС-30Т+GPU, с	183,1	174,8
Время расчета на 8 узлах кластера НКС-30Т+GPU, с		247,4

Кол-во узлов расчетной сетки	по оси X	1500
	по оси Y	700
	по оси Z	2100
	по времени	1000

ВЫВОДЫ

Для сравнения быстродействия обоих подходов проведены расчеты одинаковых сред с размером сетки $1500*700*2100$ узлов по пространству и 1000 шагов по времени.

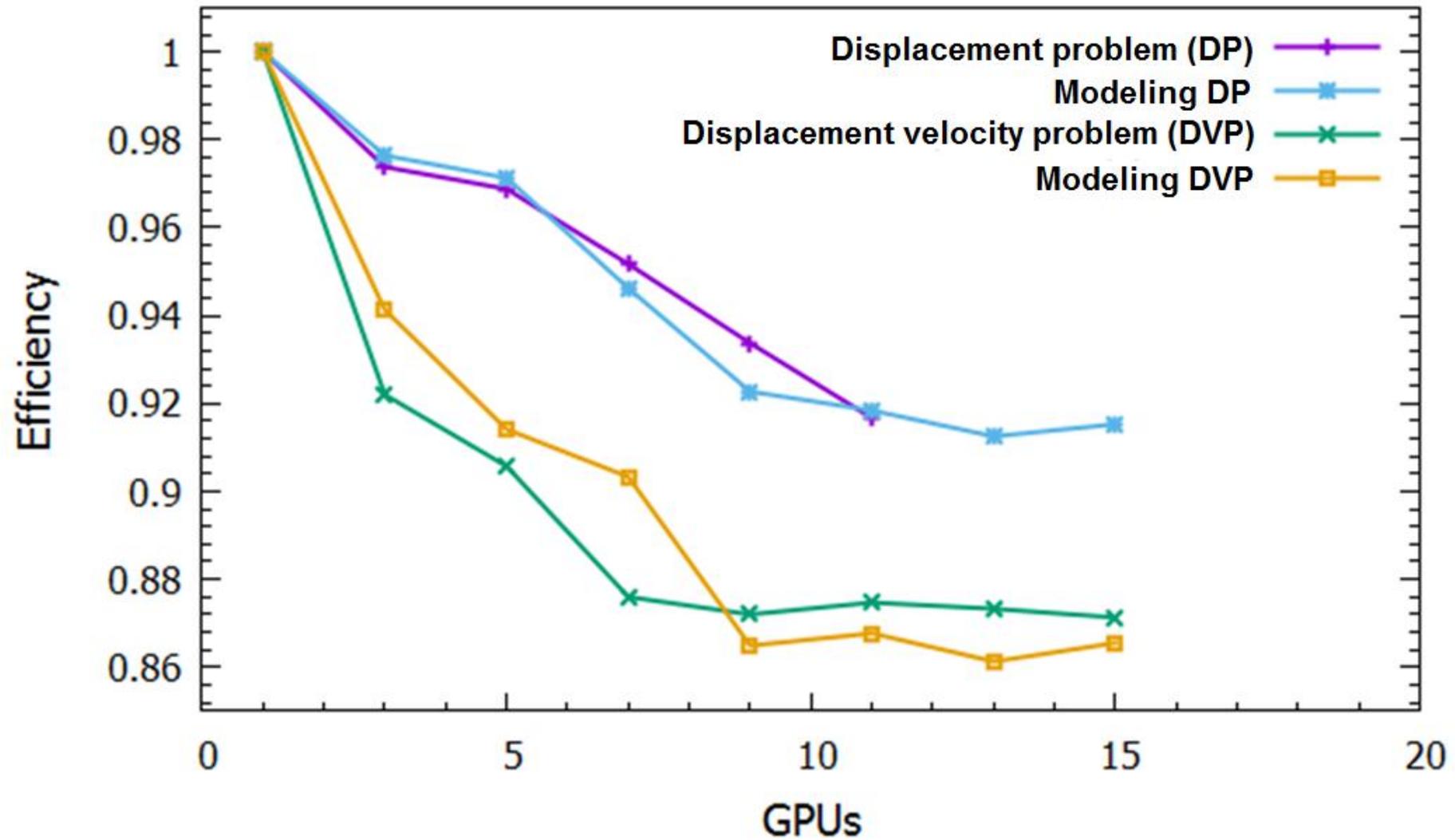
Взята сетка близкая к максимально входящей в память 45 графических карт (15 вычислительных узлов кластера НКС-30Т+GPU) при расчете скоростей смещения и напряжения.

При расчете смещений на такой же по размеру сетке используется практически в два раза меньше оперативной памяти, и он может быть проведен минимум на 8 узлах вместо 15.

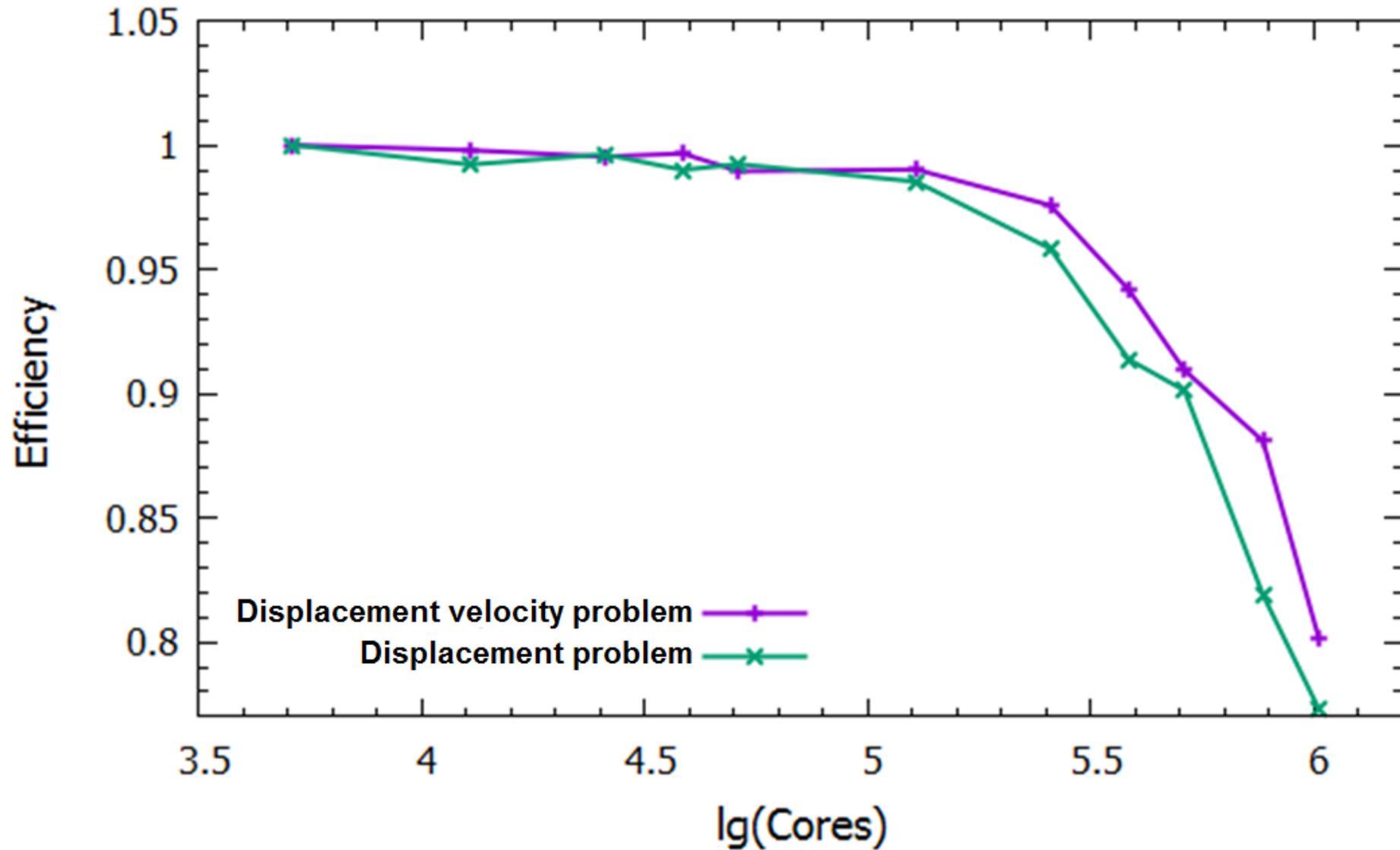
Проведенные эксперименты демонстрируют, что подход, основанный на расчете смещений, быстрее и при этом позволяет проводить расчеты для очень больших сеток, запрашивая меньшее количество свободных узлов. Это позволяет быстрее получать доступ к ним в условиях очереди на кластере, обеспечивая при этом приемлемое время расчета (несколько часов для реальной полномасштабной задачи).

Glinskii B.M., Martynov V.N., Sapetina A.F.: 3D Modeling of Seismic Wave Fields in a Medium Specific to Volcanic Structures. Yakutian Mathematical Journal. 22(3), 84–98 (2015).

ВЕРИФИКАЦИЯ МОДЕЛИ



МАСШТАБИРУЕМОСТЬ АЛГОРИТМОВ

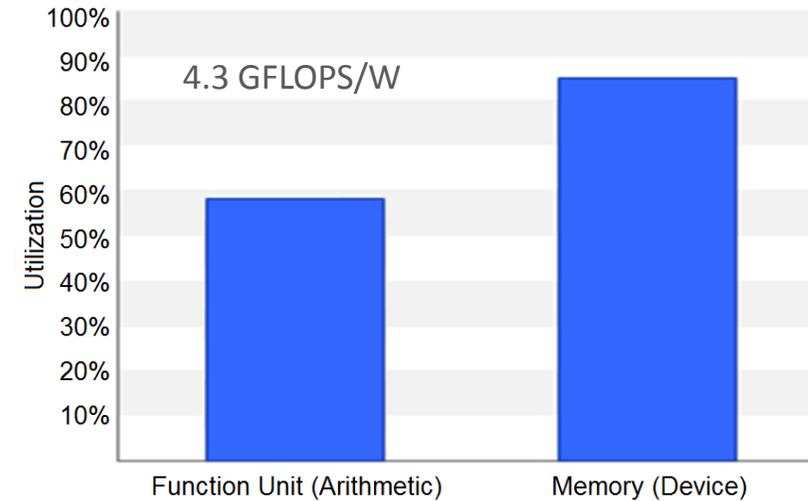
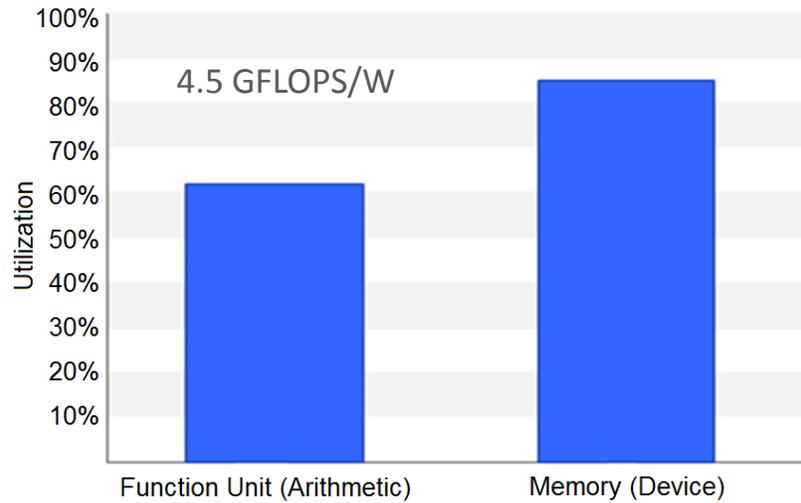


ОЦЕНКА ЭНЕРГОЭФФЕКТИВНОСТИ

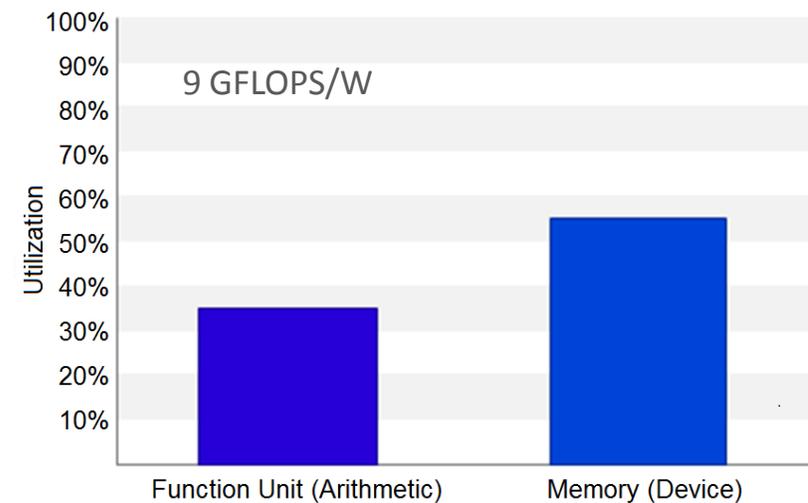
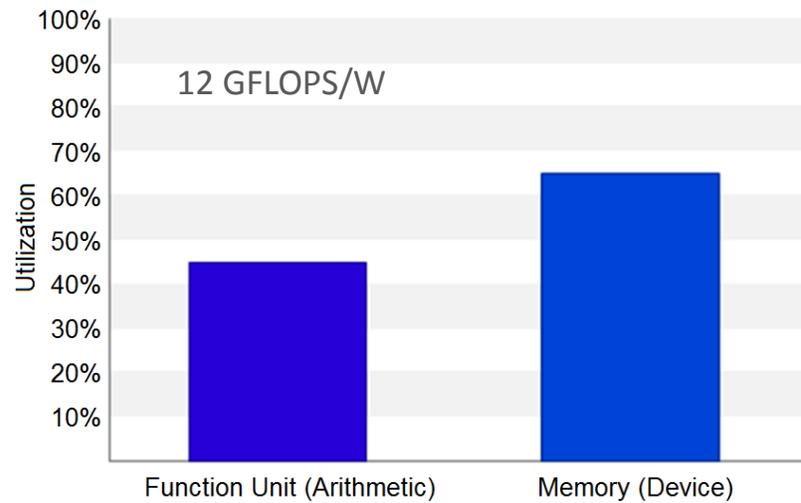
$\vec{\sigma}, \vec{u}$

\vec{U}

Tesla M2090



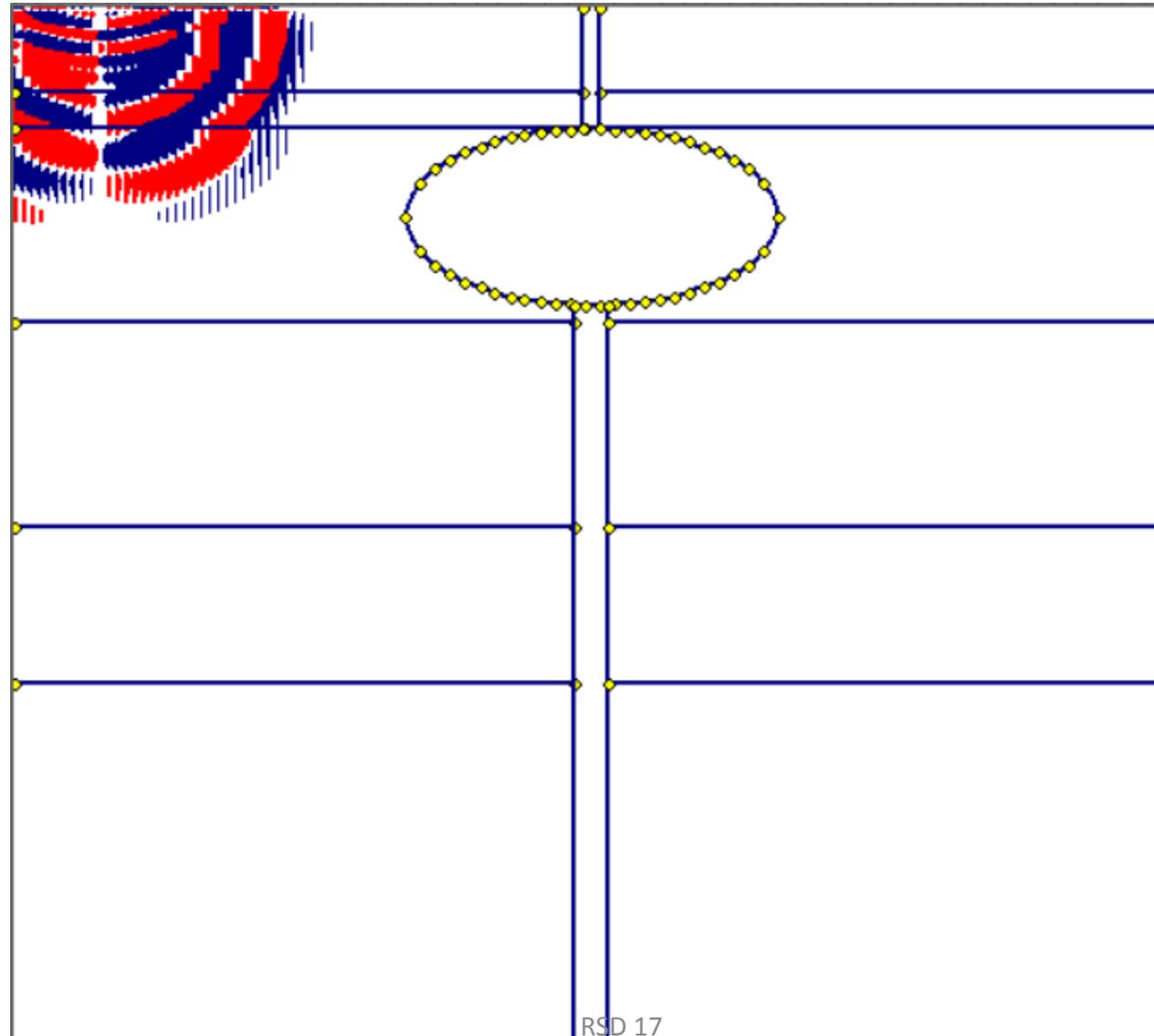
K40M



РЕЗУЛЬТАТЫ ТЕСТОВОГО МОДЕЛИРОВАНИЯ

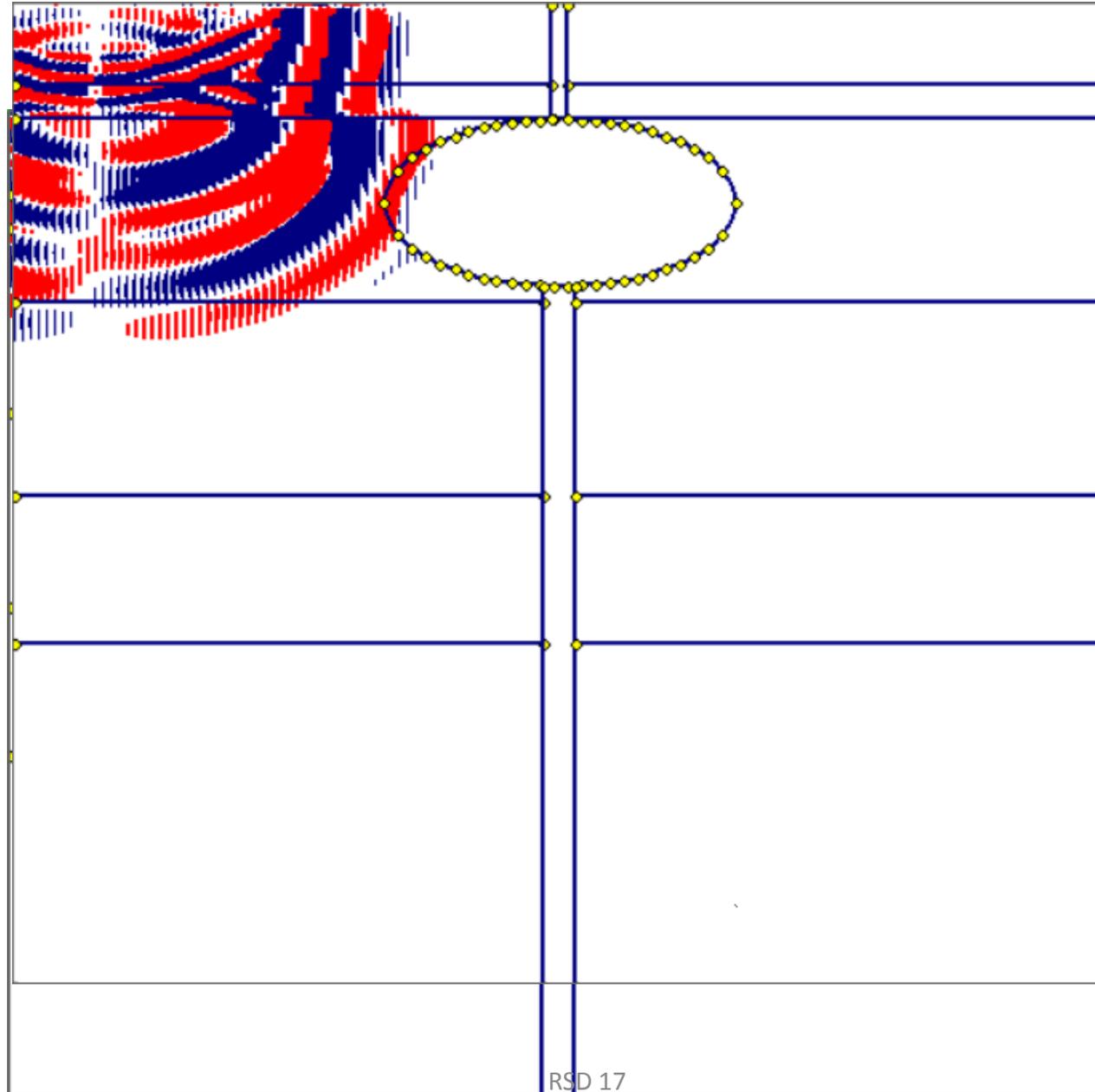
$T = 0,46 \text{ с}$

u, плоскость XZ



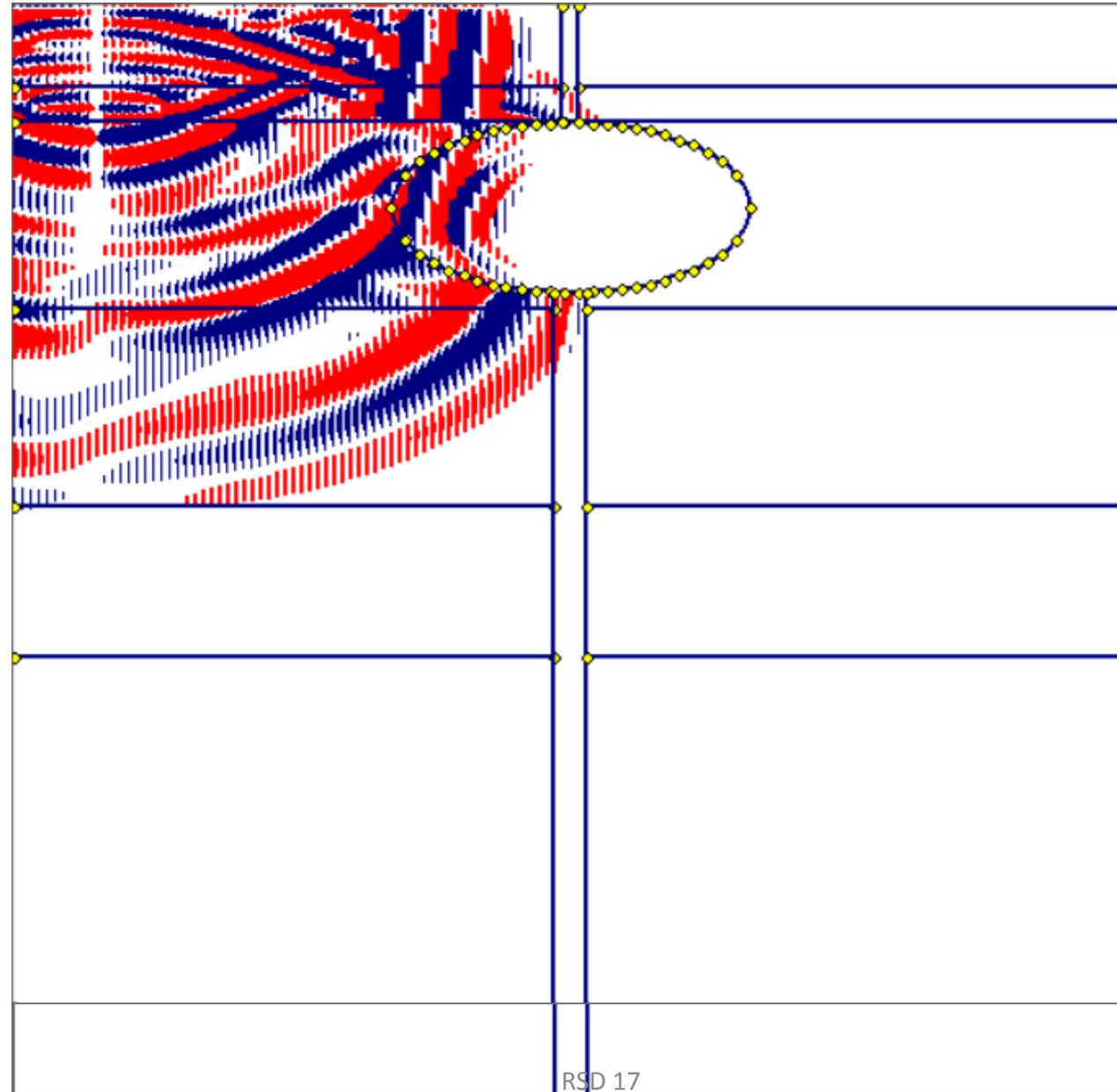
РЕЗУЛЬТАТЫ ТЕСТОВОГО МОДЕЛИРОВАНИЯ

$T = 0,69 \text{ c}$



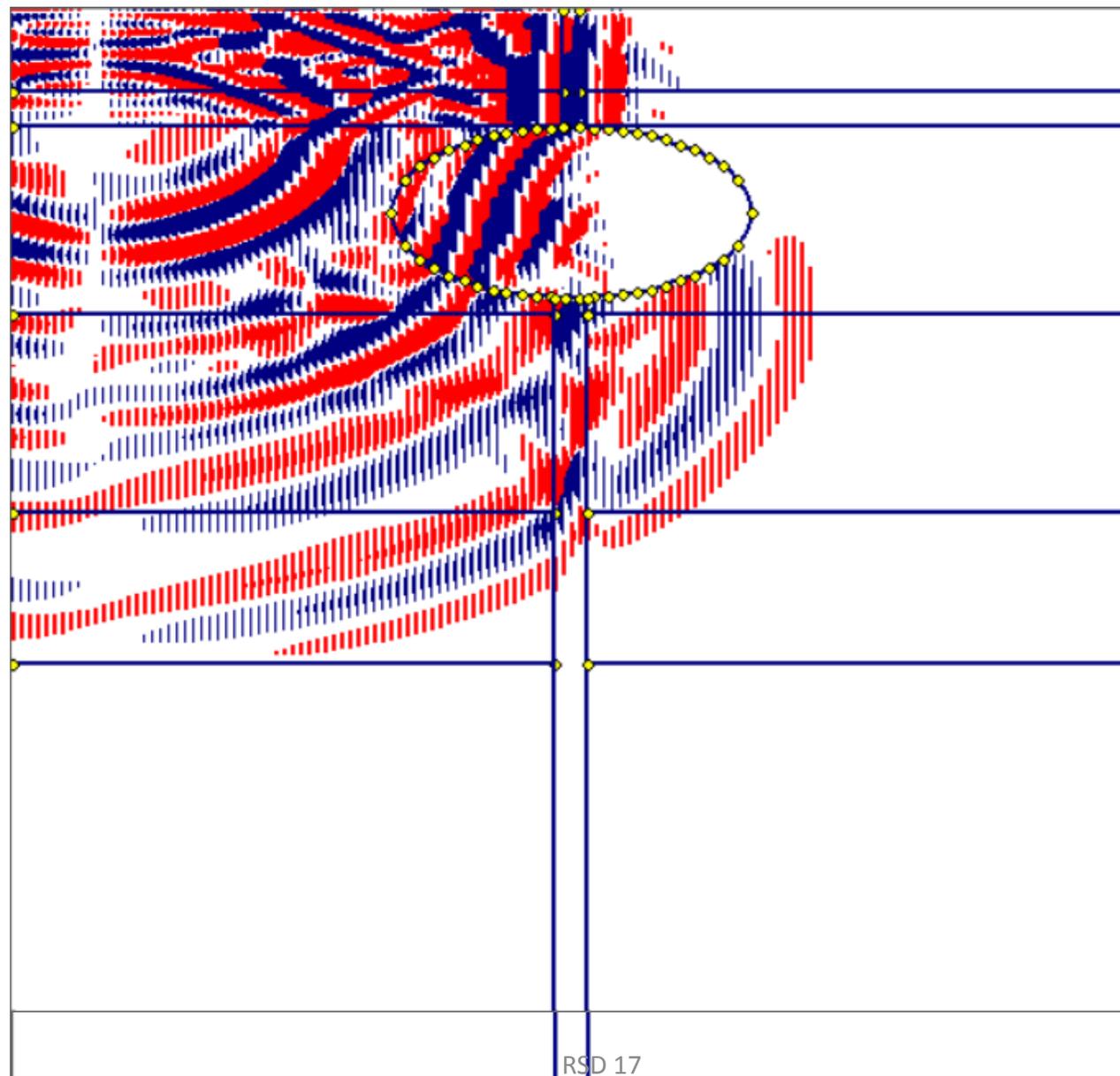
РЕЗУЛЬТАТЫ ТЕСТОВОГО МОДЕЛИРОВАНИЯ

$T = 1,15 \text{ c}$



РЕЗУЛЬТАТЫ ТЕСТОВОГО МОДЕЛИРОВАНИЯ

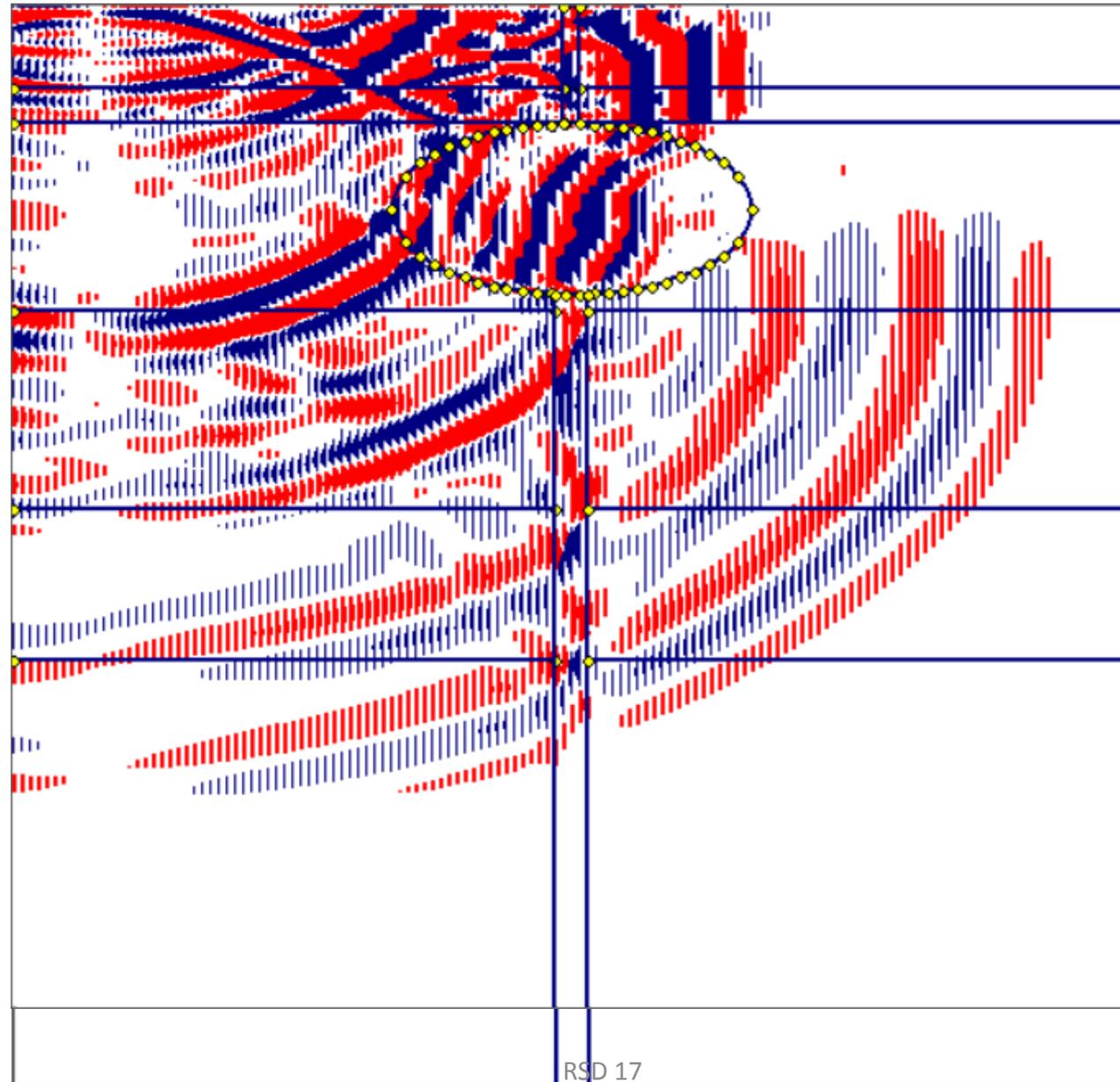
$T = 1,38 \text{ с}$



РЕЗУЛЬТАТЫ ТЕСТОВОГО МОДЕЛИРОВАНИЯ

$T = 1,61 \text{ c}$

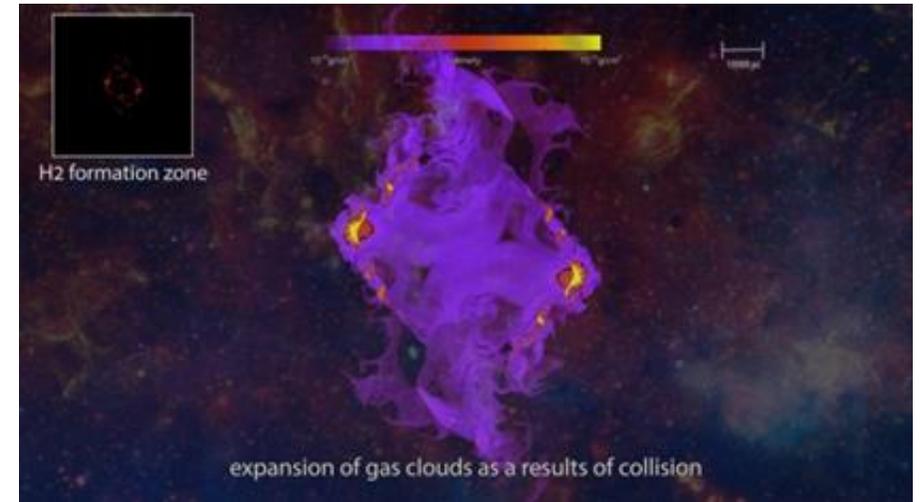
и, плоскость XZ



ОБЩИЙ ВЫВОД ПО ЗАДАЧЕ

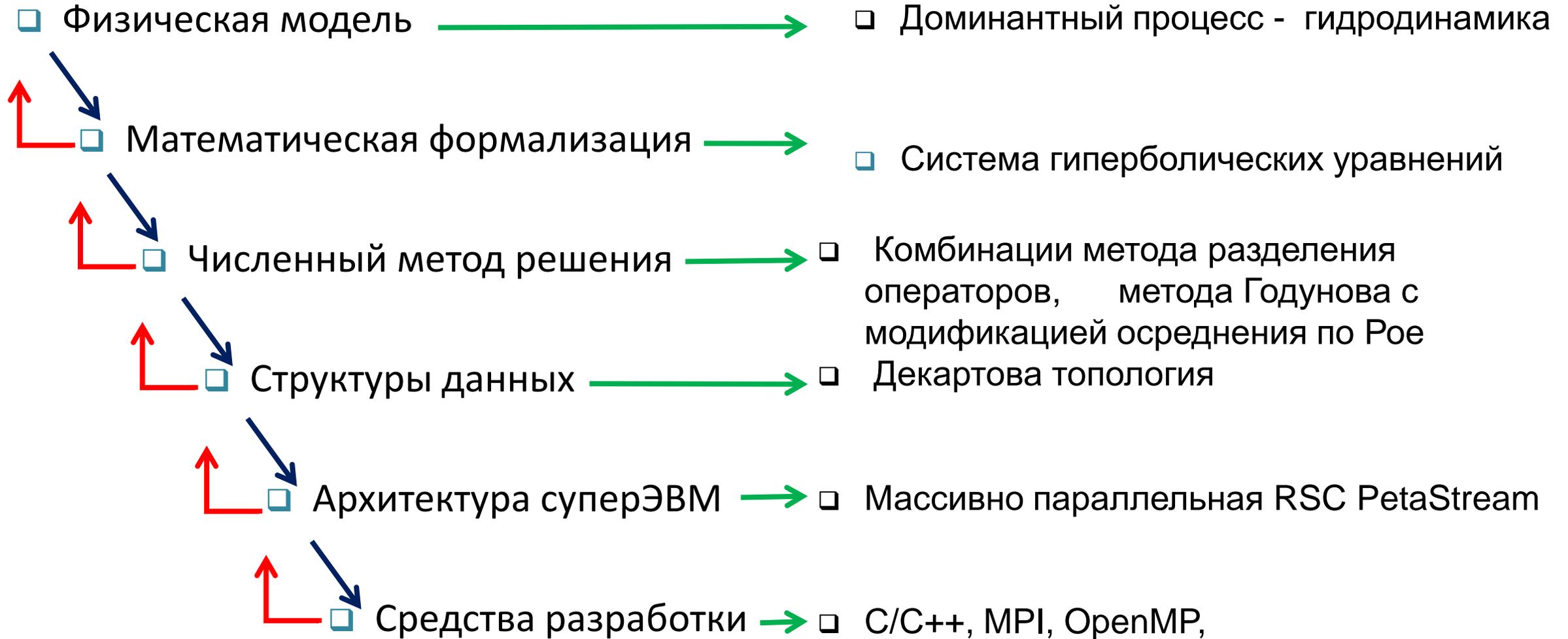
- В рамках со-дизайна проведено сравнение разработанных параллельных реализаций решения задачи динамической теории упругости, записанной в терминах скоростей смещения и напряжения и в терминах смещений для кластера, оснащенного графическими ускорителями.
- Исследовано время работы созданных параллельных программ, их масштабируемость. В том числе проведено имитационное моделирование исполнения программ на большом количестве ядер с помощью системы моделирования AGNES.
- Показано, что энергоэффективность разработанного ПО близка к 12 GFLOPS/W на ускорителе Nvidia K40M, что почти в три раза выше чем на ускорителе Nvidia Tesla 2090M.
- На основе полученных результатов можно дать рекомендации по предпочтению в использовании подхода, основанного на расчете смещений.

МНОГОКОМПОНЕНТНАЯ ГИДРОДИНАМИЧЕСКАЯ МОДЕЛЬ СТОЛКНОВЕНИЯ ГАЛАКТИК



Химодинамика столкновения галактик: начальная стадия (слева), разлет газовых облаков после столкновения и зона образования молекулярного водорода (справа). Одна галактика пролетает через другую с образованием двух газовых облаков и зоны образования молекулярного водорода после удара.

КОНЦЕПЦИЯ И РЕАЛИЗАЦИЯ СО-ДИЗАЙНА АСТРОФИЗИЧЕСКОГО МОДЕЛИРОВАНИЯ



ЧИСЛЕННЫЙ МЕТОД РЕШЕНИЯ

- Operator splitting подход (разделение на эйлеров и лагранжев этапы)
- Метод Годунова для каждого этапа
- Специальная модификация схемы Рунге для осреднения величин
- Кусочно-параболический метод на локальном шаблоне
- Быстрое преобразование Фурье
- Использование переопределенной системы уравнений (гарантия неубывания энтропии)

В частности, код AstroPhi основан на описанных методах. Для его реализации используется архитектура ускорителей Intel Xeon Phi. Для моделирования используется архитектура RSC PetaStream с 8 узлами с 64 ускорителем Intel Xeon Phi 7120D.

Kulikov I., E. Vorobyov E.: Journal of Computational Physics. 317, 318–346 (2016).

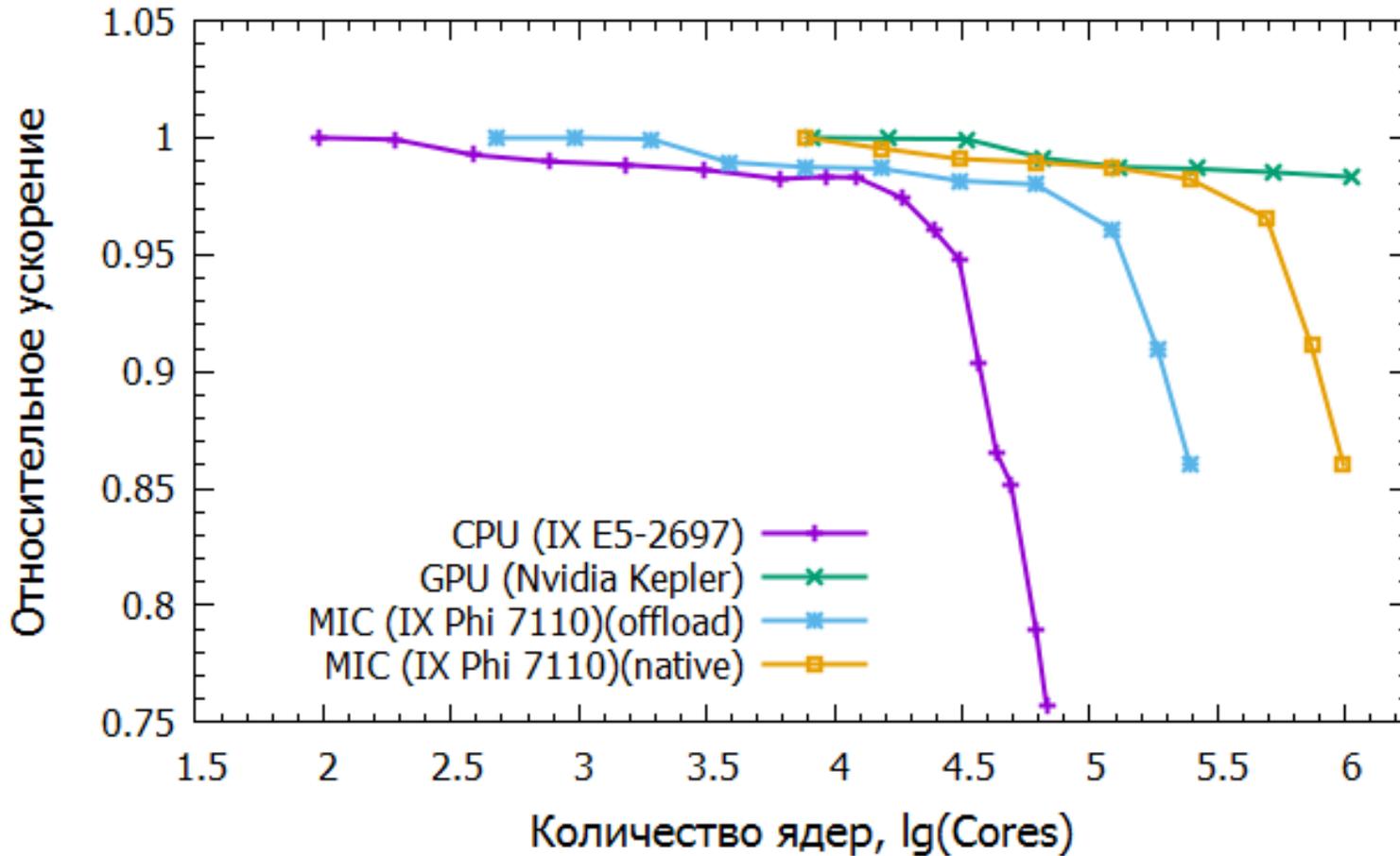
РЕЗУЛЬТАТЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ АСТРОФИЗИЧЕСКОГО КОДА

Созданы программные агенты AstroGrid. Агенты по заданным параметрам (T[Euler], T[Lagrange]) моделируют вычисления, а затем переходят к обмену данными. Агент AstroGrid отправляет сообщения двум соседям и ожидает поступления сообщений от обоих соседей. Только после этого продолжается цикл вычислений.

Во время обмена замеряются и собираются значения задержек (Λ), ведь именно задержки передачи сообщений при большом количестве узлов существенно увеличивают время выполнения алгоритма.

РЕЗУЛЬТАТЫ:

1. Время выполнения кода увеличивается незначительно (около 20 %) до 5120 вычислительных узлов;
2. Лучшую масштабируемость показал код, запущенный на узлах с ускорителями Nvidia Kepler K40 и Intel Xeon Phi (native mode).



ЭНЕРГОЭФФЕКТИВНОСТЬ

Профиль исполнения астрофизического кода на рабочем узле SMP-G7 (Intel E7-4870) ССКЦ

Name	TSelf	TSelf	TTotal	#Calls	TSelf /Call
▲ Group Application	154.396 s		157.082 s	4	38.5989 s
Process 0	38.4594 s		39.2725 s	1	38.4594 s
Process 1	38.9564 s		39.2733 s	1	38.9564 s
Process 2	38.9702 s		39.2756 s	1	38.9702 s
Process 3	38.0096 s		39.2609 s	1	38.0096 s
▷ Group MPI	2.68675 s		2.68675 s	384	6.99675e-3 s

Профиль исполнения астрофизического кода на рабочем узле G8-K40 (Intel Xeon E5-2650v2) ССКЦ

Name	TSelf	TSelf	TTotal	#Calls	TSelf /Call
▲ Group Application	44.1507e+3 s		45.0657e+3 s	16	2.75942e+3 s
Process 0	2.75706e+3 s		2.81676e+3 s	1	2.75706e+3 s
Process 1	2.7155e+3 s		2.81672e+3 s	1	2.7155e+3 s
Process 2	2.73732e+3 s		2.81674e+3 s	1	2.73732e+3 s
Process 3	2.73277e+3 s		2.81675e+3 s	1	2.73277e+3 s
Process 4	2.73852e+3 s		2.81672e+3 s	1	2.73852e+3 s
Process 5	2.70039e+3 s		2.81675e+3 s	1	2.70039e+3 s
Process 6	2.73803e+3 s		2.81674e+3 s	1	2.73803e+3 s
Process 7	2.75461e+3 s		2.81677e+3 s	1	2.75461e+3 s
Process 8	2.78787e+3 s		2.8166e+3 s	1	2.78787e+3 s
Process 9	2.78898e+3 s		2.81653e+3 s	1	2.78898e+3 s
Process 10	2.78098e+3 s		2.81635e+3 s	1	2.78098e+3 s
Process 11	2.78907e+3 s		2.81662e+3 s	1	2.78907e+3 s
Process 12	2.78067e+3 s		2.81623e+3 s	1	2.78067e+3 s
Process 13	2.79598e+3 s		2.8167e+3 s	1	2.79598e+3 s
Process 14	2.77751e+3 s		2.81614e+3 s	1	2.77751e+3 s
Process 15	2.77545e+3 s		2.81654e+3 s	1	2.77545e+3 s
▷ Group MPI	914.946 s		914.946 s	1776	515.172e-3 s

26.09.2017 RSD 17

Оптимизация численных методов астрофизического кода и использование последних версий программных продуктов из пакета Intel Parallel Studio XE 2016 Beta дали отличный результат балансировки нагрузки на ядрах процессоров. Время выполнения MPI операций составляет не более 2-3 % от общего времени счета.

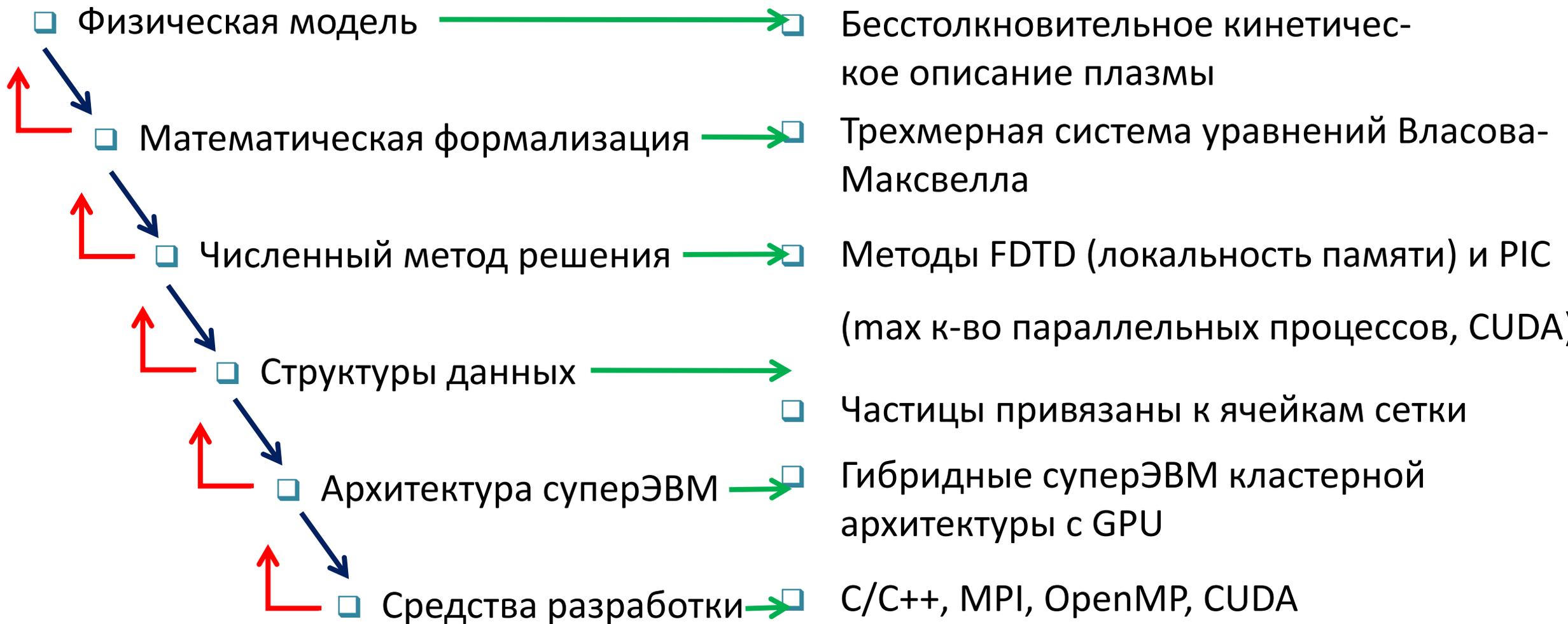
ВЗАИМОДЕЙСТВИЕ МОЩНОГО ЭЛЕКТРОННОГО ПУЧКА С ПЛАЗМОЙ.

Решается совместно с ИЯФ СО РАН

- Глобальная цель: Построение энергетического термоядерного реактора
- Используется одна из альтернативных концепций: магнитная ловушка (ГОЛ-3)
- Наблюдается термоядерная реакция слияния дейтерия
- Требуется увеличить время удержания плазмы в разогретом состоянии
- Конкретная физическая задача: **моделирование ленгмюровской турбулентности при релаксации электронного пучка в плазме**

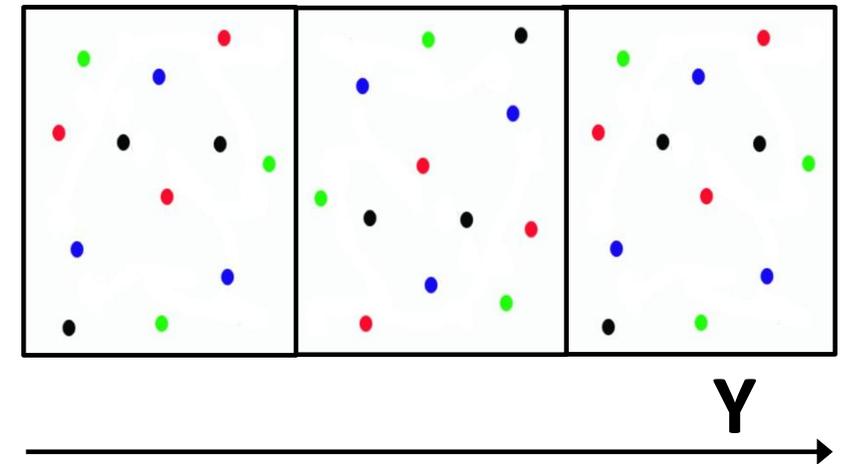


СО-ДИЗАЙН МОДЕЛИРОВАНИЯ В ФИЗИКЕ ПЛАЗМЫ



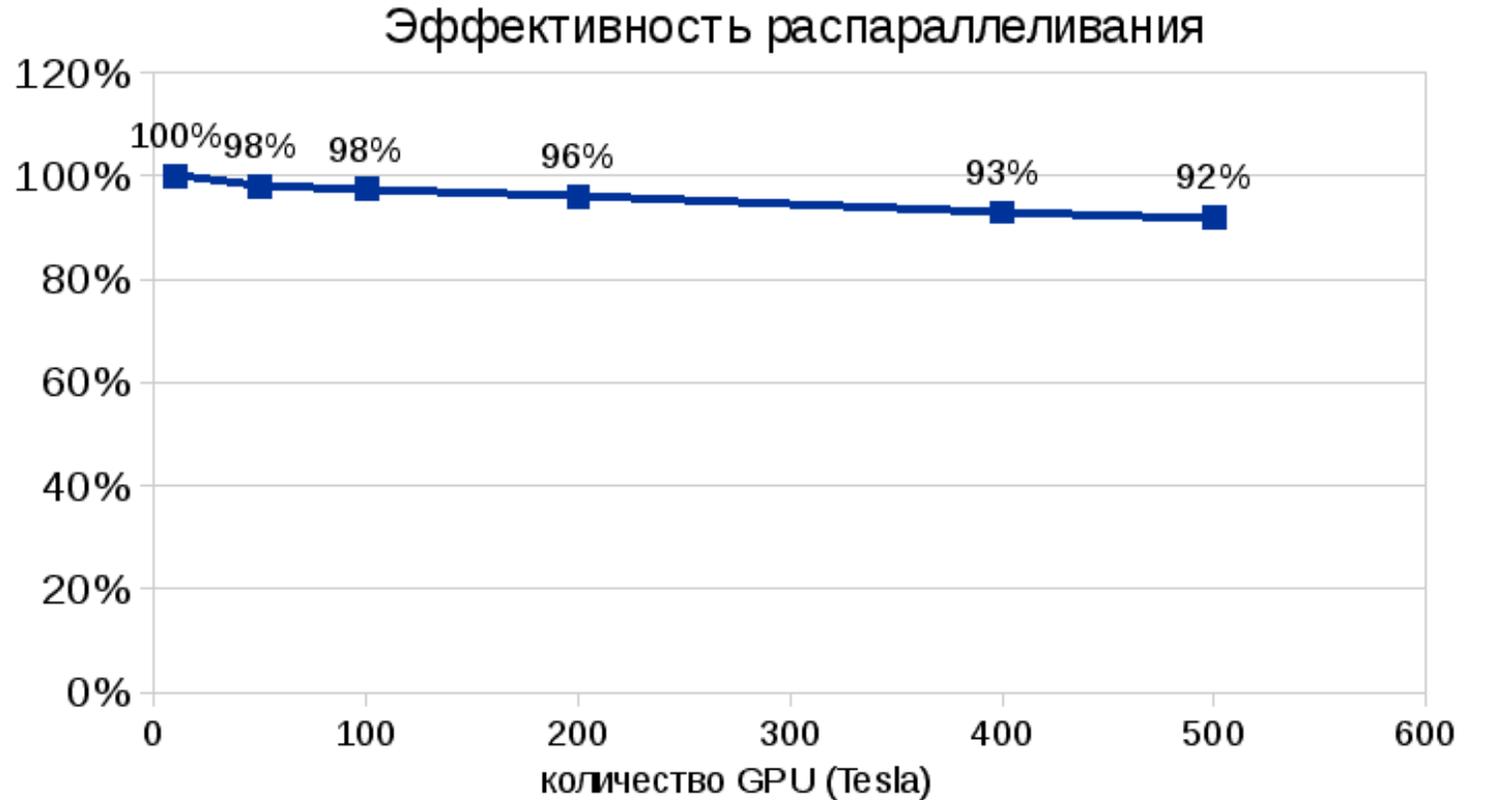
СТРАТЕГИЯ РАСПАРАЛЛЕЛИВАНИЯ

- Уравнения Максвелла решаются методом FDTD, кинетическое уравнение – методом частиц в ячейках
- Расчетная область делится на N_{MAX} подобластей вдоль координаты Y для решения уравнений Максвелла (пучок летит вдоль X)
- Для каждой подобласти назначается группа из M процессорных элементов (или GPU, или ускорителей Intel Xeon Phi)
- Далее частицы каждой подобласти разделяются дополнительно между M процессорными элементами (частицы, принадлежащие разным ПЭ на схеме показаны разными цветами)



ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

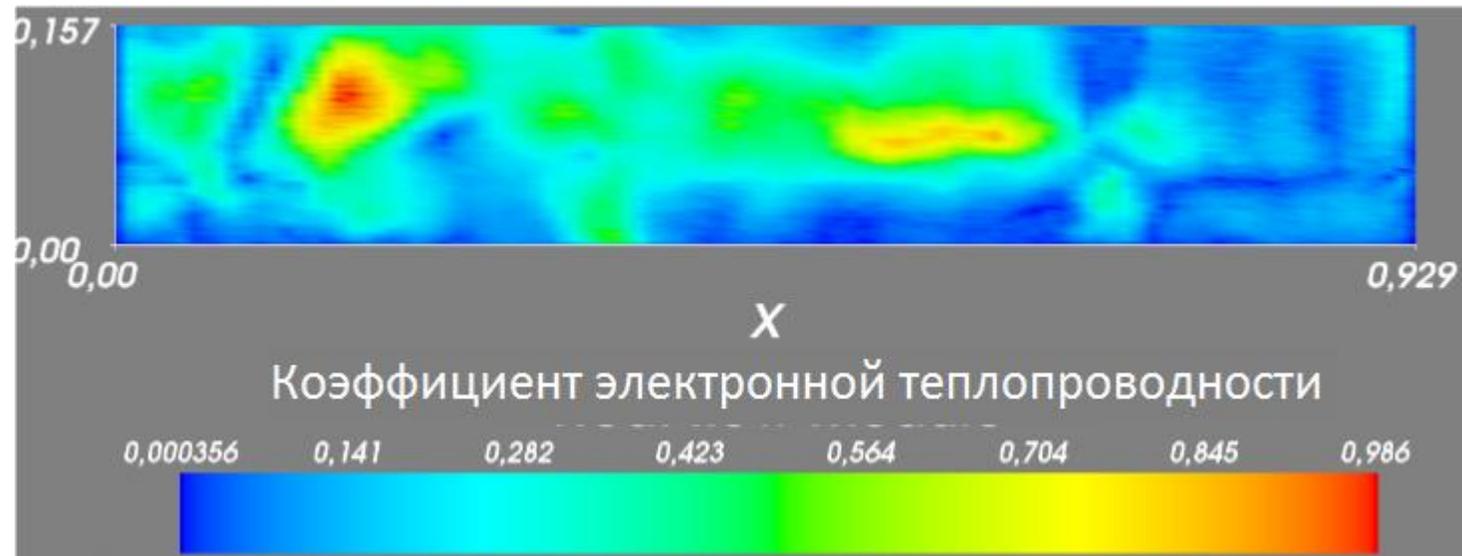
На суперкомпьютере “Ломоносов” (НИВЦ МГУ) проведен тестовый расчет по моделированию ленгмюровской турбулентности с использованием **160 миллионов модельных частиц** на 500 GPU.



Эффективность распараллеливания составляет более 90% для 500 узлов.

Lotov K.V., Timofeev I.V., Mesyats E.A., Snytnikov A.V., V.A.Vshivkov «Note on quantitatively correct simulations of the kinetic beam-plasma instability», Physics of Plasmas, Feb. 2015

МОДЕЛИРОВАНИЕ ТЕПЛООВОГО ПОТОКА В ПЛАЗМЕ ПОСЛЕ РЕЛАКСАЦИИ ПУЧКА



После релаксации мощного электронного пучка коэффициент электронной теплопроводности в некоторых частях расчетной области уменьшается на 2-3 порядка, что качественно соответствует физике процесса (значения на рисунке нормированы на **начальное значение** коэффициента электронной теплопроводности)

РЕЗУЛЬТАТЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ КОДА ФИЗИКИ ПЛАЗМЫ

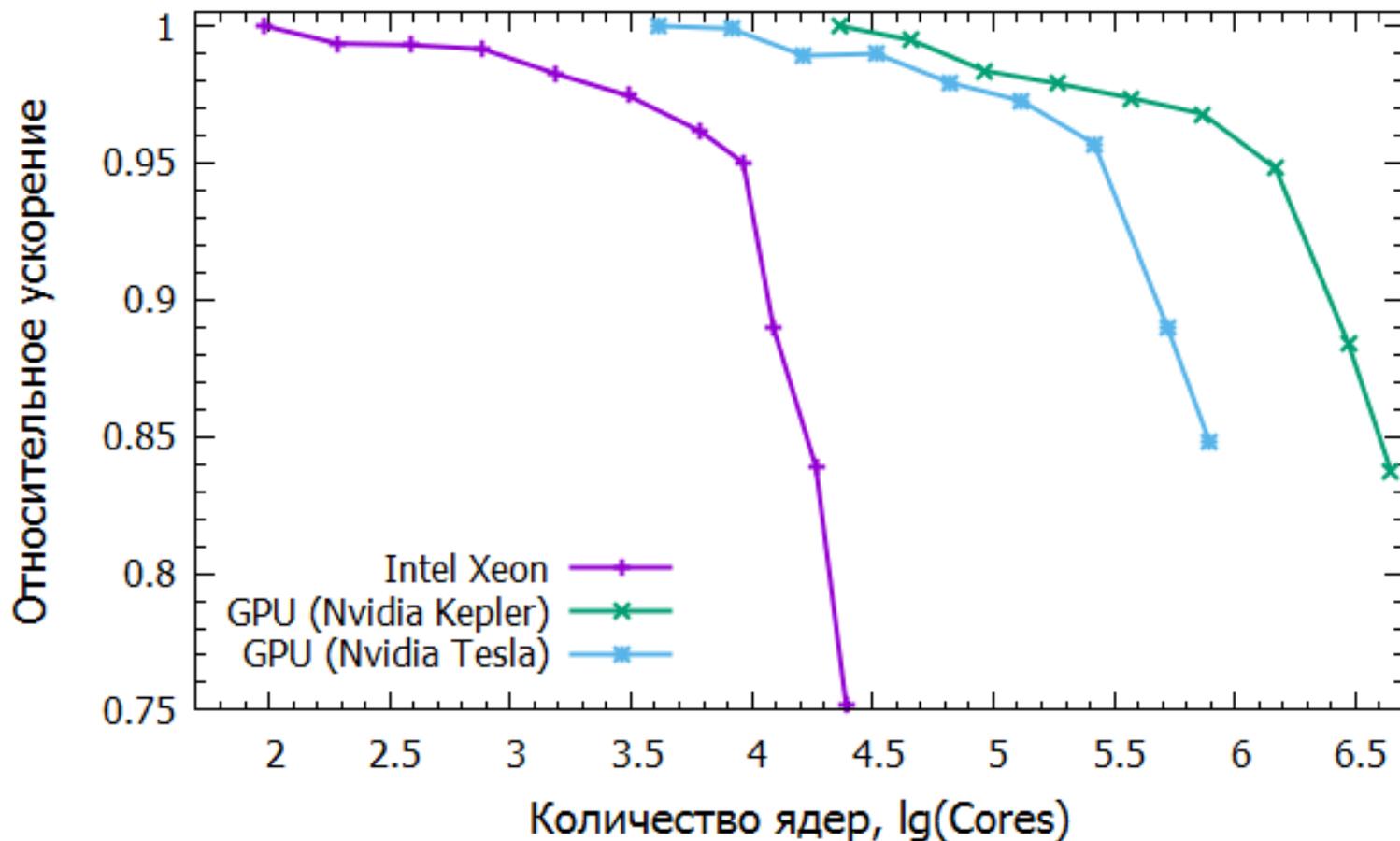
Созданы программные агенты PlasmaGrid. Агенты по заданным параметрам (T[Maxvell]) моделируют вычисления, а затем переходят к обмену данными. Так как перед следующей итерацией цикла агенты должны суммировать данные всех остальных (MPI_Allreduce) вводятся дополнительно агенты PlasmaBar.

Агенты PlasmaBar собирают промежуточные результаты со всех PlasmaGrid и передают обобщенные данные обратно, что является сигналом к продолжению вычислений.

РЕЗУЛЬТАТЫ:

1. На этапе формирования полной матрицы текущей плотности тока и заряда необходим обмен значениями «все-со-всеми». Заметно, что эти обмены существенно снижают эффективность алгоритма.

2. Код физики плазмы хорошо масштабируется на вычислительных узлах с ускорителями Nvidia Kepler K40 GPUs;



ПРОФИЛЬ ИСПОЛНЕНИЯ КОДА ФИЗИКИ ПЛАЗМЫ НА G8-K40 NODES ССКЦ

Name	TSelf	TSelf	TTotal	#Calls	TSelf/Call
▲ Group Application	389.82 s		422.531 s	30	12.994 s
Process 0	12.5916 s		13.9837 s	1	12.5916 s
Process 1	12.941 s		14.2177 s	1	12.941 s
Process 2	12.9323 s		14.2493 s	1	12.9323 s
Process 3	12.5263 s		14.037 s	1	12.5263 s
Process 4	13.4026 s		14.132 s	1	13.4026 s
Process 5	13.6495 s		14.1396 s	1	13.6495 s
Process 6	12.9546 s		14.0106 s	1	12.9546 s
Process 7	13.4088 s		14.1386 s	1	13.4088 s
Process 8	13.5501 s		14.1408 s	1	13.5501 s
Process 9	13.0014 s		14.0147 s	1	13.0014 s
Process 10	12.6144 s		14.1758 s	1	12.6144 s
Process 11	12.828 s		14.1234 s	1	12.828 s
Process 12	12.3164 s		14.0278 s	1	12.3164 s
Process 13	12.6281 s		14.159 s	1	12.6281 s
Process 14	13.1659 s		14.0234 s	1	13.1659 s
Process 15	13.3675 s		14.0839 s	1	13.3675 s
Process 16	13.1959 s		14.1117 s	1	13.1959 s
Process 17	13.3358 s		14.168 s	1	13.3358 s
Process 18	13.4274 s		14.1031 s	1	13.4274 s
Process 19	13.2678 s		14.1091 s	1	13.2678 s
Process 20	12.3406 s		14.0329 s	1	12.3406 s
Process 21	12.8743 s		14.08 s	1	12.8743 s
Process 22	12.3167 s		13.9653 s	1	12.3167 s
Process 23	12.314 s		14.0282 s	1	12.314 s
Process 24	13.3187 s		14.1198 s	1	13.3187 s
Process 25	13.4332 s		14.023 s	1	13.4332 s
Process 26	12.8908 s		13.992 s	1	12.8908 s
Process 27	13.1831 s		14.1562 s	1	13.1831 s
Process 28	13.09 s		14.0077 s	1	13.09 s
Process 29	12.9528 s		13.9768 s	1	12.9528 s
▶ Group MPI	32.7115 s		32.7115 s	9600660	3.40721e-6 s

Со-дизайн решения задачи, подбор численных методов, использование новейших CUDA 7.0 и методов оптимизации вместе с Intel Parallel Studio привело к хорошей балансировки нагрузки, время выполнения MPI операций составляет не больше 10 %

ЗАКЛЮЧЕНИЕ

- Предлагается методика разработки алгоритмов и программного обеспечения для суперкомпьютеров экзафлопсного класса, содержащая три связанных этапа: со-дизайн; имитационное моделирование поведения алгоритма при масштабировании; оценка энергоэффективности алгоритма.
- Подход апробирован на трех задачах: сейсмика, астрофизики и физики плазмы.
- Исследование масштабируемости алгоритмов проводилось с применением имитационной системы AGNES, путем его отображения на гипотетическую архитектуру суперкомпьютера.
- Данный подход позволил получить для указанных задач распараллеленные коды с хорошей балансировкой и минимальным временем выполнения MPI операций.
- Показано, что алгоритмы решения данных задач для заданной архитектуры суперкомпьютера эффективно могут использовать около 1 млн вычислительных ядер.
- Для дальнейшего увеличения вычислительных ядер необходимо радикальное переработка вычислительных методов и алгоритмов решения задач.

**Спасибо
за внимание!**

www2.sscs.ru