
**Реализация итерационных методов решения
СЛАУ для гибридных вычислительных
систем и их использование при
моделировании диссипативных процессов в
задачах физики плазмы**

П.А. Кучугов

Г.А. Багдасаров, А.С. Болдарев, О.Г. Ольховская, Ю.А. Повещенко

Институт прикладной математики им. М.В. Келдыша РАН

**Параллельные вычислительные технологии
(ПАВТ'2016)**

г. Архангельск, Архангельская обл., РФ,
28 марта -1 апреля 2016



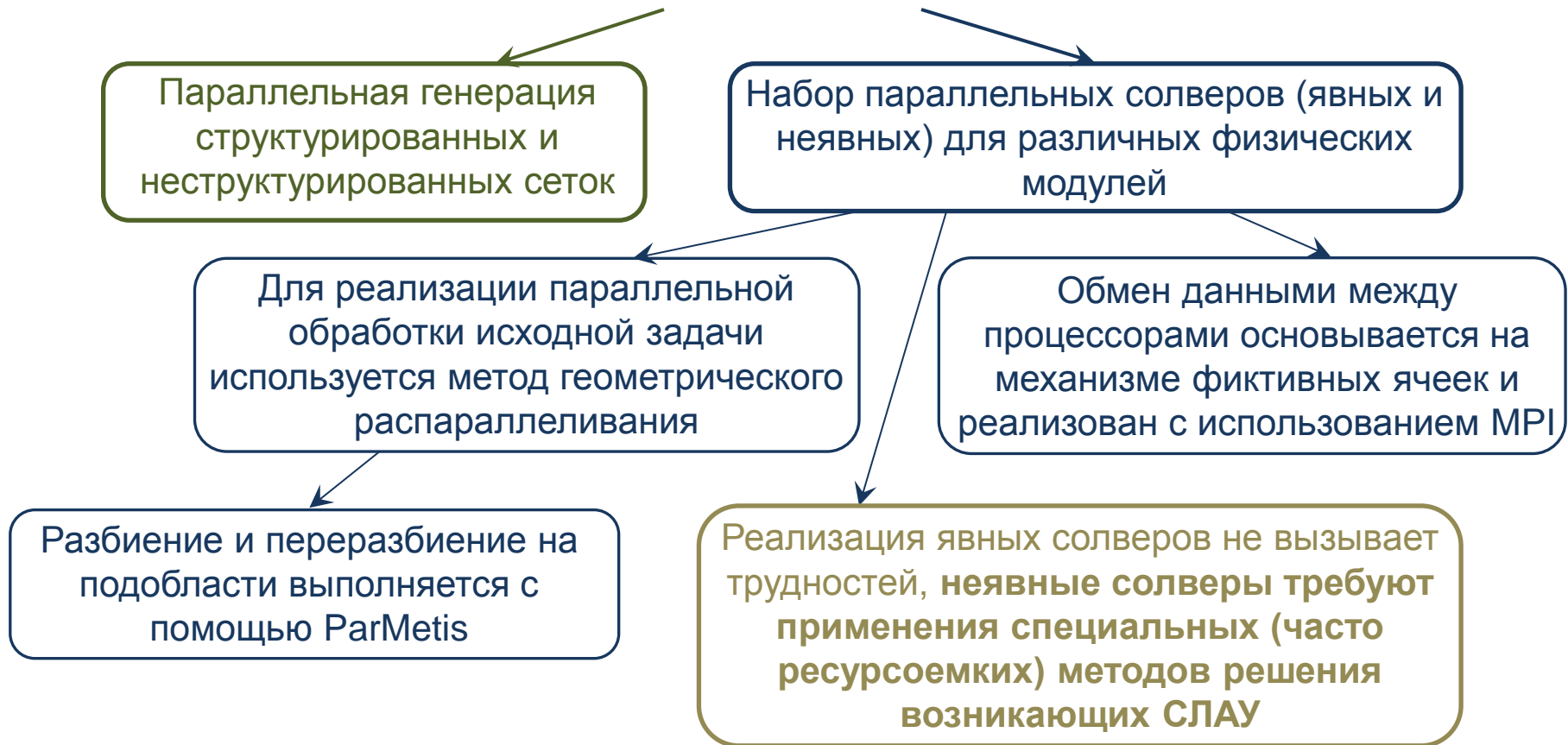
- При моделировании задач, в которых возникают системы линейных алгебраических уравнений (СЛАУ), в большинстве случаев основная часть вычислительного времени уходит на их решение
- Реализации решателей, работающих с распределёнными матрицами и задействующих ресурсы графических со-процессоров, представлены не достаточно широко
- Работа со сторонними библиотеками в большинстве случаев подразумевает необходимость преобразования форматов данных программы к форматам данных, используемым внутри библиотек, что увеличивает время решения СЛАУ



Программная реализация MARPLE3D



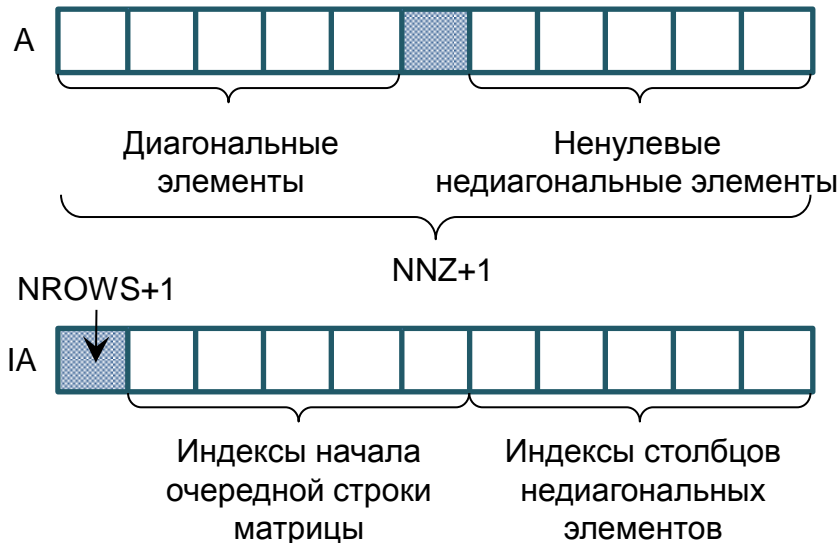
Пакет прикладных программ MARPLE3D (C++)



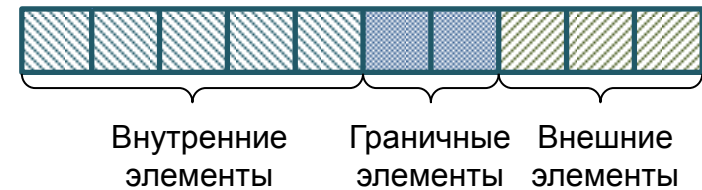
Решение распределённых СЛАУ (CPU)

Для решения СЛАУ на CPU используется библиотека Aztec*

Формат хранения локальной разреженной матрицы - MSR



Формат хранения распределённого вектора



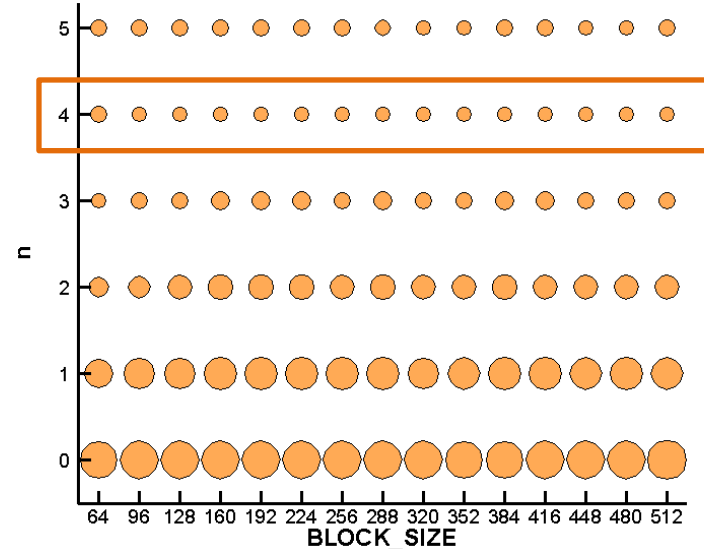
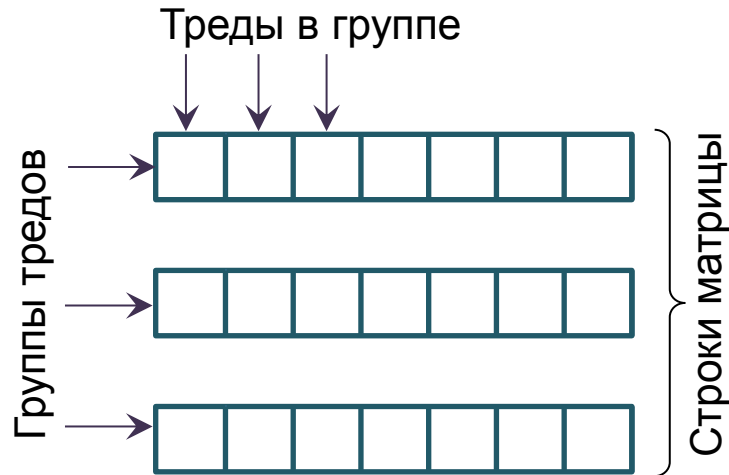
Используется решатель **BiCGStab** с 3-х шаговым предобуславливателем по методу Гаусса-Зейделя, применяемым к локальной матрице

В наиболее общем случае при каждом вычислении произведения матрицы на вектор необходимо обновлять внешние элементы вектора

*R.S. Tuminaro, M. Heroux, S. Hutchinson, J.N. Shadid, Official Aztec User's Guide. Version 2.1, SAND99-8801J, 1999.

Решение распределённых СЛАУ (CPU+GPU)

Умножение локальной разреженной матрицы на вектор



$groupId = tid/NT$, $lane = tid\%NT$,
 где NT – число тредов,
 обрабатывающих одну строку матрицы

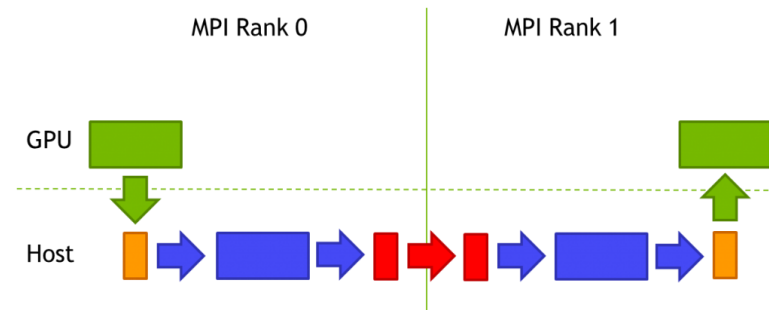
Зависимость времени вычисления SpMV от
 размера блока (BLOCK_SIZE) и числа тредов
 в группе ($NT=2^n$) на примере матрицы для
 задачи о распространении ЭМГ-волны

- Наименьшее время вычисления SpMV соответствует размеру группы тредов $NT=16$ для большей части размеров блока
- С учетом отсутствия необходимости преобразования формата матрицы, время вычисления SpMV в данной реализации меньше, чем при использовании CUB, CUSP или CUSPARSE*

*См. дополнительные слайды

Решение распределённых СЛАУ (CPU+GPU)

- Для большинства векторных операций (скалярное произведение, умножение вектора на скаляр и др.) используются функции CUBLAS
- Для отдельных групп повторяющихся векторных операций реализованы объединённые ядра с целью уменьшить их общее количество
- Наиболее общая схема обмена данными между несколькими GPU-устройствами – необходимо минимизировать число таких обменов
- Выбор предобуславливателя с высокой степенью параллелизма



Параметры для оценки реализации решателя

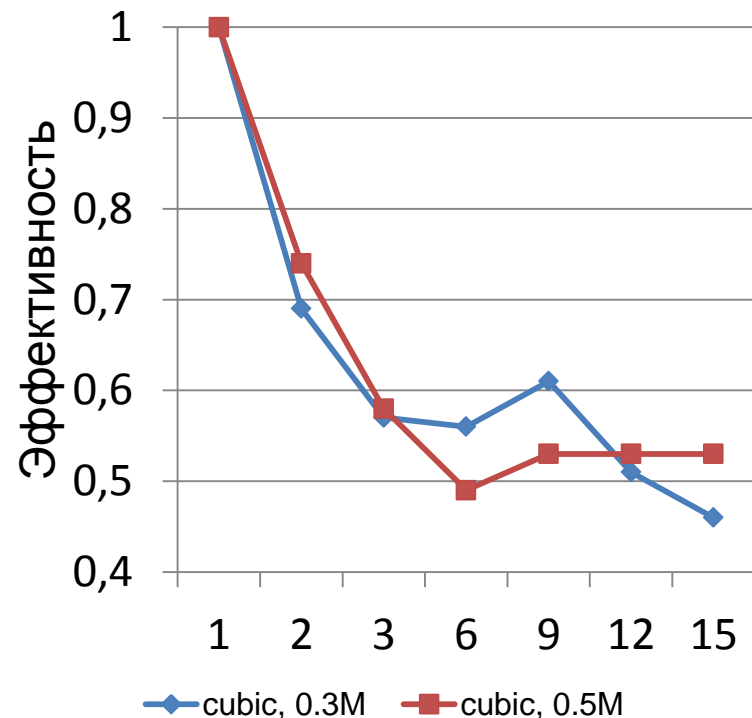
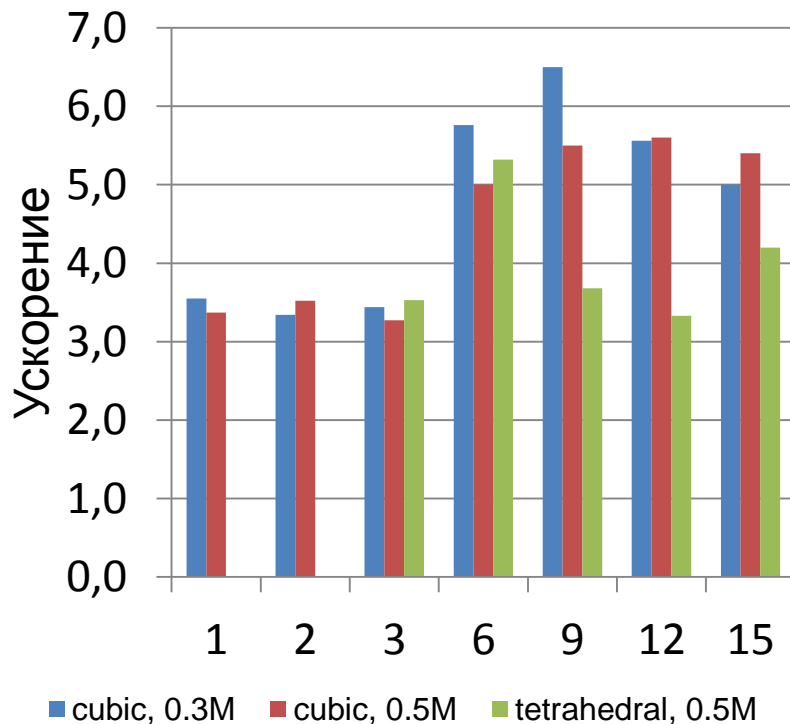
- Ускорение за счёт использования GPU: $S = t_N / T_N$
- Эффективность при фиксированном размере задачи: $E = T_1 / (NT_N)$

где N – число CPU-процессов, t_N – время выполнения одной итерации решателя N CPU-процессами, T_N – то же, но с дополнительным использованием ресурсов одного GPU-устройства для каждого CPU-процесса

Примеры моделирования и результаты*

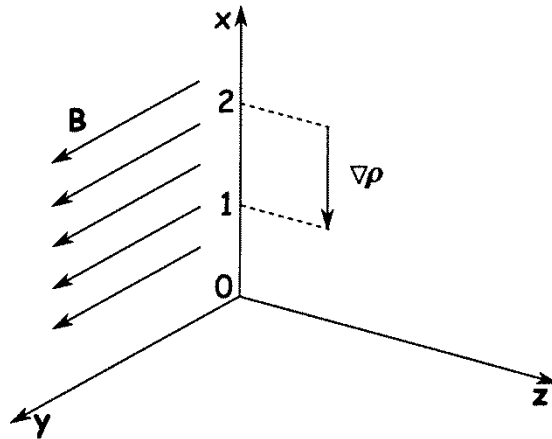
$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial x_i} \left(\kappa_0 T^\alpha \frac{\partial T}{\partial x_i} \right), \quad x_i \in [0,3], \quad \alpha = 2.0, \quad \kappa_0 = 0.5, \quad s_0 = 0.5,$$

$$D = 5.0, \quad \tau = 2 \cdot 10^{-4}, \quad t_{\max} = 0.4$$



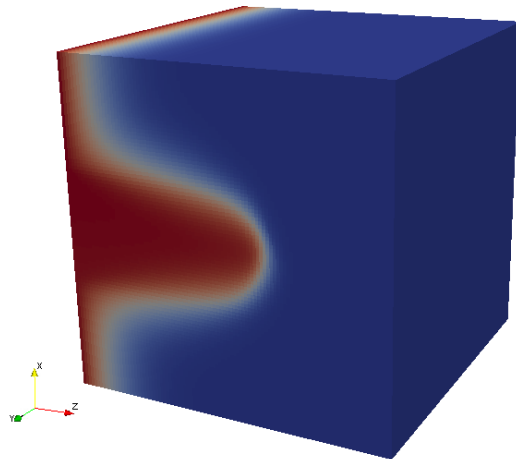
*См. дополнительные слайды для уточнения постановки задачи

Примеры моделирования и результаты*



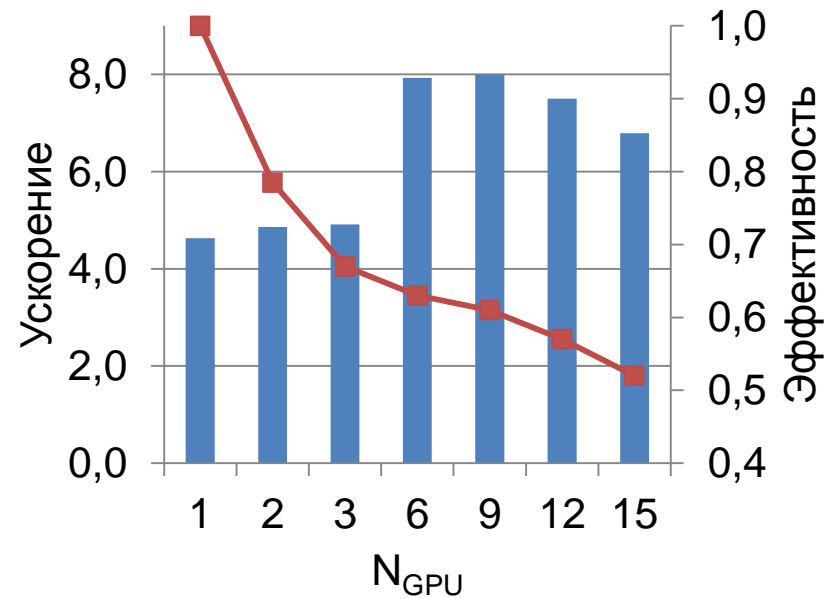
$$\frac{\partial B}{\partial t} + \kappa B \frac{\partial B}{\partial z} = D \frac{\partial^2 B}{\partial z^2}, \quad B \equiv B_y,$$

$$D = 1/4\pi, \quad \kappa = 8.25 \cdot 10^{-18} / \rho$$



Результат моделирования в момент времени $t=1.0$

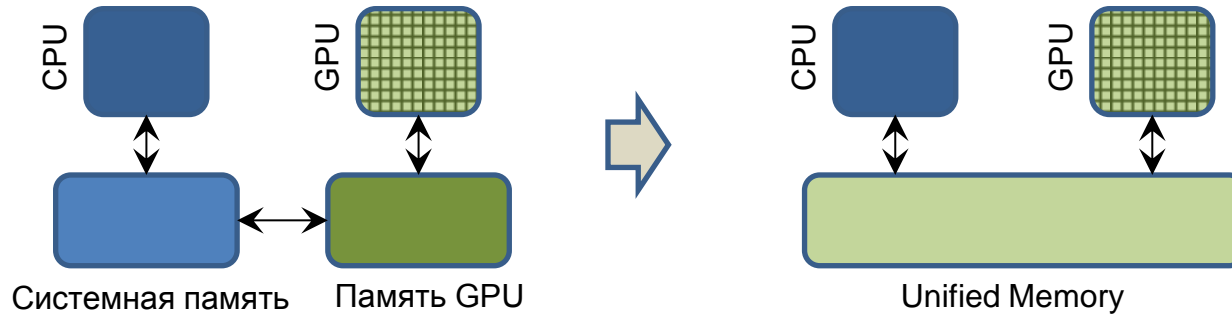
Ускорение и эффективность при фиксированном размере задачи



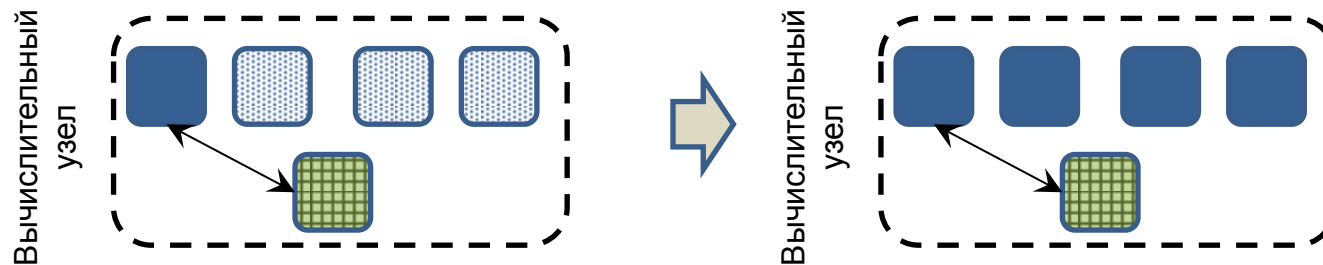
*См. дополнительные слайды для уточнения постановки задачи

Планируемые исследования и доработки

- Перенос большего количества вычислений на GPU, например, процедуры заполнения матрицы СЛАУ
- Использование Unified Memory (CUDA 6.0 и выше) для автоматизации (и оптимизации) обменов между GPU и CPU



- Балансировка вычислений между CPUs и GPUs в рамках одного узла для его полноценной загрузки





- Представленная реализация метода BiCGStab с полиномиальным предобуславливателем позволяет добиться ускорения одной итерации решателя в диапазоне от 3.5 до 8 раз за счёт использования ресурсов графических сопроцессоров.
- Пиковые значения ускорения достигаются в случае, когда на одно GPU-устройство приходится от 30 до 50 тыс. строк исходной матрицы СЛАУ

Спасибо за внимание!





Дополнительные слайды



Постановка задачи о тепловой волне

Граничные условия

Начальные условия

$$T(0, t) = \left[\frac{\alpha D}{\kappa_0} (Dt + s_0) \right]^{1/\alpha}, \quad t > 0, \quad T(x_i, 0) = \begin{cases} \left[\frac{\alpha D}{\kappa_0} (s_0 - x_i) \right]^{1/\alpha}, & x_i \leq 0, \\ 0, & x_i > 0 \end{cases}$$

Аналитическое решение: тепловая волна распространяется с постоянной скоростью

$$T(x_i, t) = \begin{cases} \left[\frac{\alpha D}{\kappa_0} (Dt + s_0 - x_i) \right]^{1/\alpha}, & x_i \leq s_0 + Dt, \\ 0, & x_i > s_0 + Dt, \end{cases}$$

где s_0 свободный параметр, D , скорость распространения тепловой волны, α и κ_0 , произвольные коэффициенты

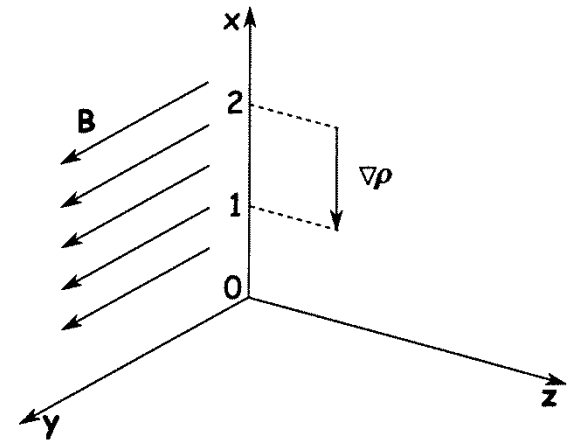
А.А. Самарский, И.М. Соболев. Примеры численного расчёта температурных волн, 3(4):945–970, 1963.

Постановка задачи о распространении ЭМГ-волны



Распределение плотности,
использовавшееся для вычисления κ

$$\rho(x) = \begin{cases} 2\rho_0, & x \leq 1, \\ 2\rho_0/x, & 1 < x \leq 2, \\ \rho_0, & 2 < x \leq 3, \end{cases} \quad \text{где } \rho_0 = 10^{-18}$$



Начальные и граничные условия

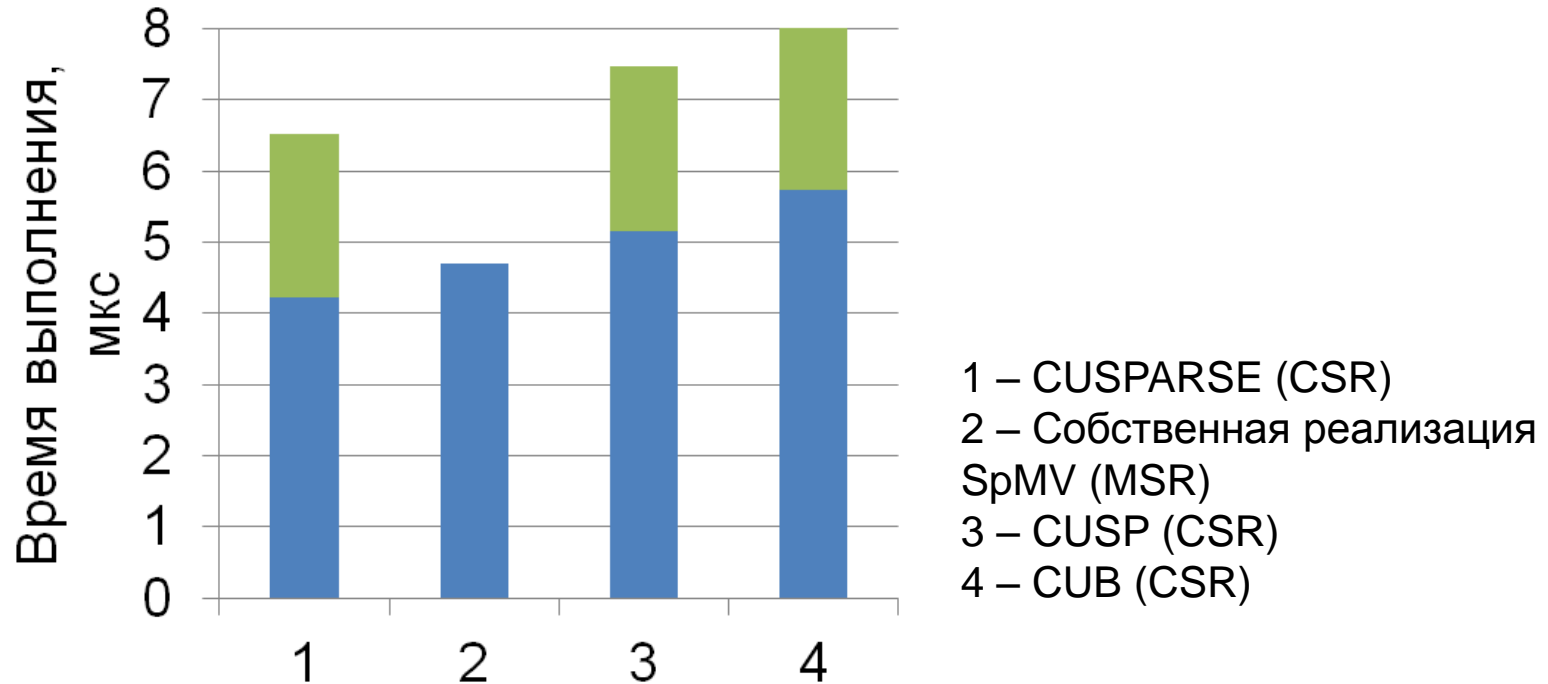
$$B(z_{\min}, t) = 1/\sqrt{4\pi}, \quad B(z_{\max}, t) = 0, \quad \left. \frac{\partial B}{\partial s} \right|_o = 0, \quad s = x | y$$

Аналитическое решение

$$B(z, t) = (B_0/2) \{1 - \tanh[(\kappa B_0/4D)(z - ut)]\}, \quad u = \kappa B_0/2$$

A.S. Kingsep, Y.V. Mokhov, and K.V. Chukbar. Nonlinear skin phenomena in plasma. In Nonlinear and turbulent processes in physics. Volume 1. Nonlinear effects in plasma physics, astrophysics and elementary particle theory, 1983.

Различные реализации SpMV

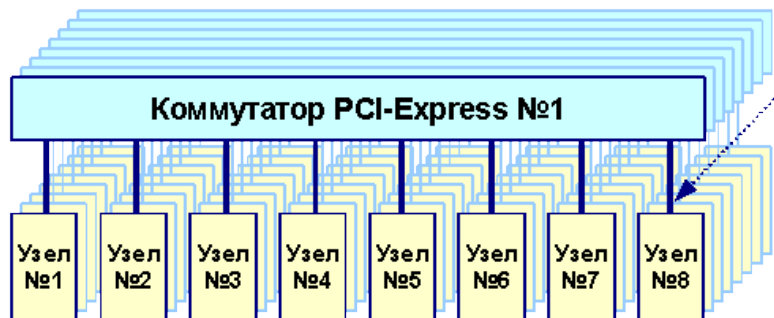


Время вычисления (мкс) произведения разреженной матрицы на вектор (SpMV) на примере матрицы для задачи о распространении ЭМГ-волны в различных реализациях, а также время преобразования формата MSR, используемого в основной программе, к какому-либо стандартному формату хранения разреженной матрицы (в данном случае к CSR)

Спецификация вычислительного кластера К-100



Скорость до 700 Мбайт/с
Латентность ~ 12 мкс
Время выдачи слова ~ 70нс
Время чтения слова ~ 2.5мкс



64 узла, каждый из которых оснащён:

12-ю вычислительными ядрами (2x CPU Intel Xeon X5670, 2.93 ГГц)

3-мя видеокартами nVidia Fermi C2050 (448 cores, 1.15 ГГц)

DDR3 SD RAM 96 Гб

Вычислительный узел

2 x CPU
Intel Xeon X5670

6 ядер на процессоре (12 на узле)
Кэш 12Мбайт
Частота ядра 2.93 ГГц

DDR3
SDRAM

DDR3 SDRAM 96 Гбайт

3 x GPU
Fermi C2050

448 ядер CUDA (1344 на узле)
Частота ядра 1.15 ГГц
2,5 Гб памяти GDDR5 (7,5 на узле)
Частота памяти 1.5 ГГц
Пропускная способность памяти 144 Гб/с

Гибридный вычислительный кластер К-100. <http://www.kiam.ru/MVS/resources/k100.html>

Спасибо за внимание!