



Библиотека XAMG для решения систем уравнений со многими правыми частями

Б.И. Краснопольский, А.В. Медведев

НИИ механики МГУ

krasnopolsky@imec.msu.ru



Содержание

План доклада:

1. При чем тут CFD?..
2. Общие принципы дизайна кода библиотеки
3. Математические методы
4. Особенности распараллеливания
5. Некоторые результаты тестирования производительности
6. Планы, проблемы...



Содержание

План доклада:

1. При чем тут CFD?..
2. Общие принципы дизайна кода библиотеки
3. Математические методы
4. Особенности распараллеливания
5. Некоторые результаты тестирования производительности
6. Планы, проблемы...



Приложение к CFD

Предпосылки:

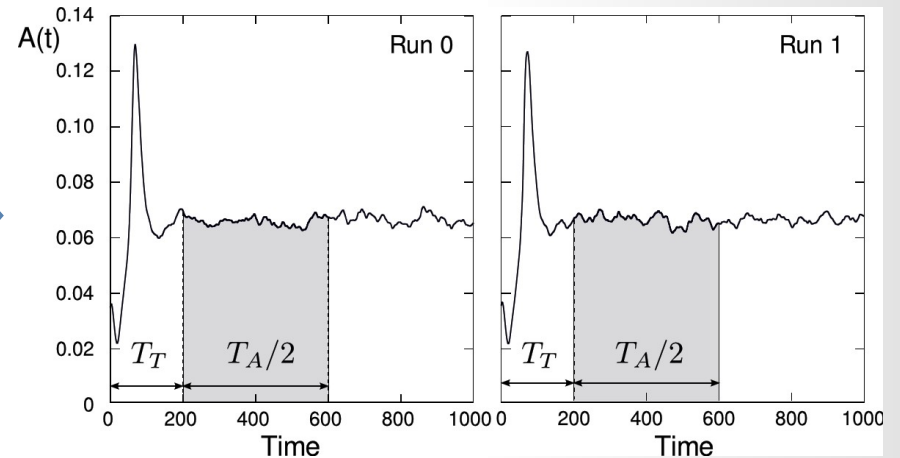
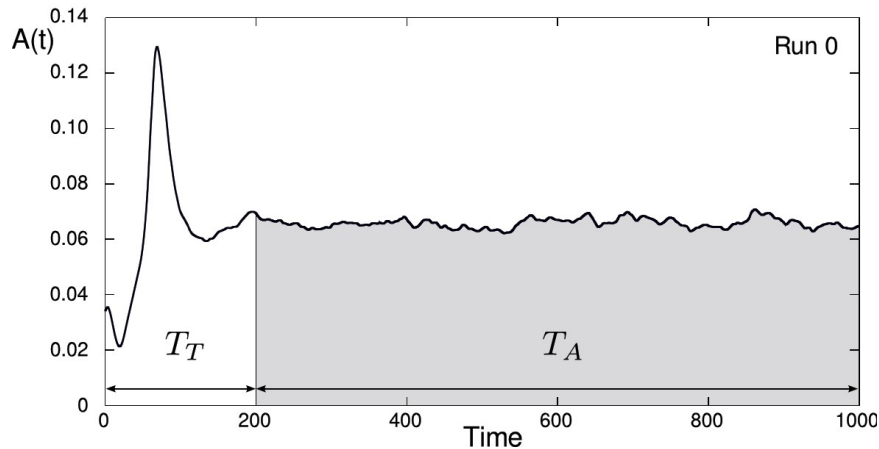
- * При расчете несжимаемых течений существенное время тратится на решение уравнения для давления
- * Для моделирования турбулентных течений в рамках вихреразрешающих методов требуется расчет большого количества шагов по времени для набора статистики
- * Ряд численных методов предполагает сохранение матрицы для давления в ходе всего расчета
 - ↳ матрица зависит только от расчетной сетки



Осреднение по ансамблю

Основная идея:

- Использовать сочетание осреднения по времени и по ансамблю реализаций для “распараллеливания” по времени
- В рамках одного расчета моделировать несколько реализаций





Преимущества и недостатки

Плюсы:

- * Решение СЛАУ со многими правыми частями
 - ↪ Потенциал ускорения по памяти до 2.5 раз
 - ↪ Улучшение масштабируемости
- * Потенциальный выигрыш в точности осреднения

Минусы:

- * Увеличение суммарного времени интегрирования задачи
- * Подход оправдан не для всех задач
- * Необходима реализация методов решения СЛАУ со многими правыми частями



Содержание

План доклада:

1. При чем тут CFD?..
2. Общие принципы дизайна кода библиотеки
3. Математические методы
4. Особенности распараллеливания
5. Некоторые результаты тестирования производительности
6. Планы, проблемы...



Базовые требования

Базовые требования к коду:

- * Ориентация на решение “эллиптических” СЛАУ
- * Фокус только на solve-часть методов
- * СЛАУ $\sim 10^8$ неизвестных с 1-64 правыми частями
- * Хорошая масштабируемость до хотя бы $\sim 10^4$ ядер
 - ↳ гибридная реализация
- * Использование графических ускорителей
- * Векторизация всех базовых операций



Дизайн кода XAMG

Основные особенности:

- * Разработан “с нуля” в 2019-2020 г.
- * C++11, yaml, C API
- * Header-style, широкое использование шаблонов
 - ↳ как следствие: в ряде случаев – длительная компиляция
- * Гибридная модель MPI + POSIX shared memory
 - ↳ как следствие: только linux/unix
- * CSR формат хранения разреженных матриц
- * Использование *hypre* для построения иерархии матриц MG



Содержание

План доклада:

1. При чем тут CFD?..
2. Общие принципы дизайна кода библиотеки
- 3. Математические методы**
4. Особенности распараллеливания
5. Некоторые результаты тестирования производительности
6. Планы, проблемы...



Математические методы (1)

Итерационные методы:

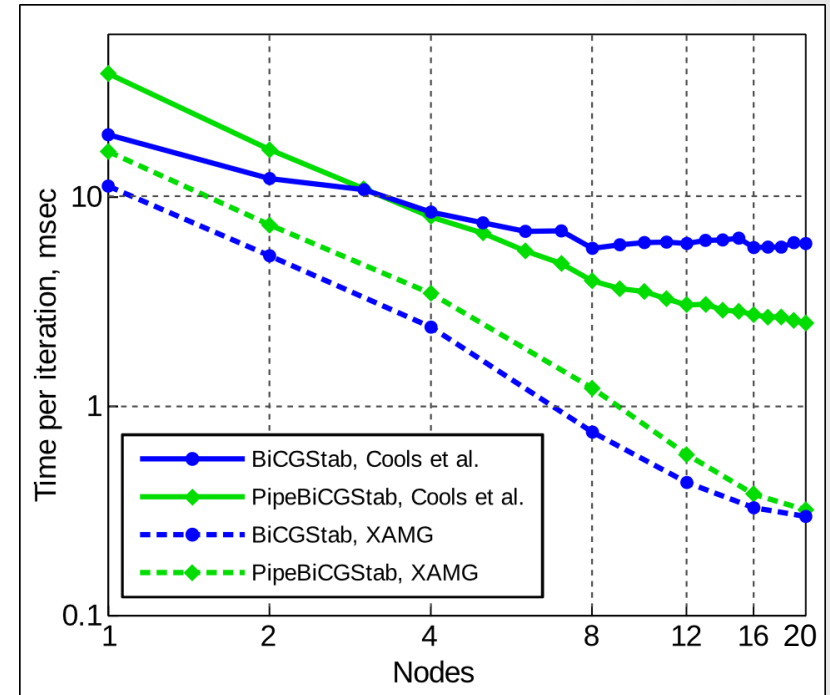
- * Псевдо-блочная реализация
- * “Стандартный” набор итерационных методов
 - ↪ Итерационные методы подпространства Крылова (CG, BiCGStab)
 - ↪ Классический алгебраический многосеточный метод
 - ↪ Методы Якоби, Гаусса-Зейделя, итераций Чебышева



Математические методы (2)

И несколько “нестандартных” методов:

- * Pipelined BiCGStab, Reordered BiCGStab
 - ↪ акцент на сокрытие глобальных коммуникаций вычислениями
 - ↪ есть много нюансов с неблокирующими глобальными операциями



1. B. Krasnopolsky, Revisiting Performance of BiCGStab Methods for Solving Systems with Multiple Right-Hand Sides // CAMWA, 2020
2. A. Medvedev, Towards benchmarking the asynchronous progress of non-blocking MPI point-to-point and collective operations // Advances in Parallel Computing, 2020.
3. B. Krasnopolsky, Predicting Performance of Classical and Modified BiCGStab Iterative Methods // Advances in Parallel Computing, 2020.



Математические методы (3)

И несколько “нестандартных” методов:

- * “Объединенные” формулировки итерационных методов (*fused*, *merged*)
 - ↪ FLOPs не играют роли
 - ↪ Сокращение операций чтения/записи за счет максимального переиспользования уже зачитанных из памяти данных

Метод	m=1	m=4	m=16
BiCGStab	7%	9%	10%
IBiCGStab	18%	23%	25%
PipeBiCGStab	21%	25%	27%
PBiCGStab	3%	6%	7%
RBiCGStab	10%	12%	14%
PPipeBiCGStab	21%	24%	26%

Выигрыш времени расчета по сравнению с базовым вариантом, тесты на Ломоносов-2



Содержание

План доклада:

1. При чем тут CFD?..
2. Общие принципы дизайна кода библиотеки
3. Математические методы
- 4. Особенности распараллеливания**
5. Некоторые результаты тестирования производительности
6. Планы, проблемы...



Распараллеливание

Особенности параллельной реализации:

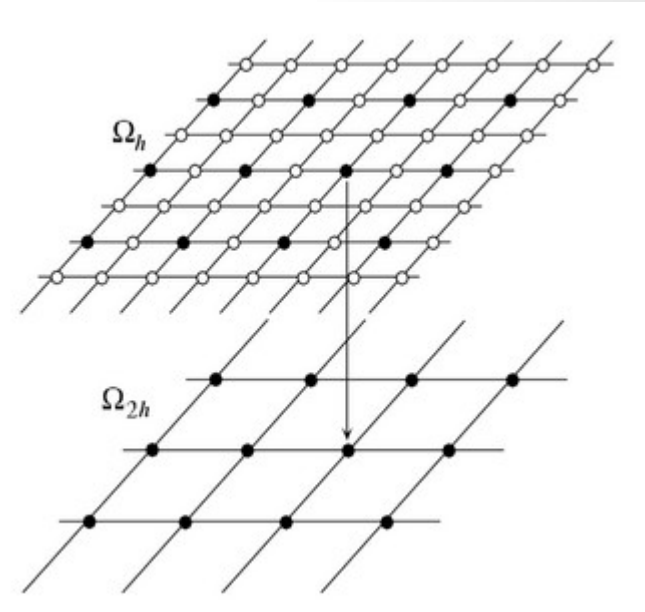
- * Трехуровневое иерархическое распараллеливание
 - ↪ node / numa / core (соответствие архитектуре железа)
- * Гибридная модель MPI+POSIX shared memory
 - ↪ “ленивое” сопряжение с вычислительными кодами
 - ↪ запуск обычной MPI программы
 - ↪ выделение локальных коммуникационных буферов в общей памяти, обмен в пределах узла через локальные буфера
 - ↪ ровно один коммуникационный процесс на узел



Локальные оптимизации кода (1)

Ряд специфических оптимизаций:

- 1) Флаги статуса “ноль/не ноль” для матриц и векторов
- 2) Смешанная точность для вычислений с плавающей точкой
- 3) Поуровневое построение иерархии матриц MG



Trottenberg et. al, Multigrid, 2000



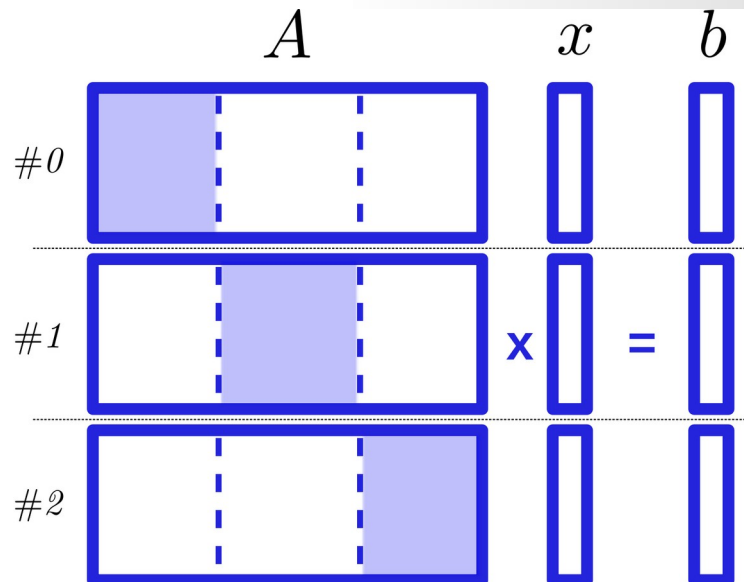
Локальные оптимизации кода (2)

Ряд специфических оптимизаций:

4) Сжатие целочисленных типов

Оценка ускорения на основе затрат памяти:

- * Смешанная точность для MG: 10-20%
- * Сжатие целочисленных типов: 10-15%





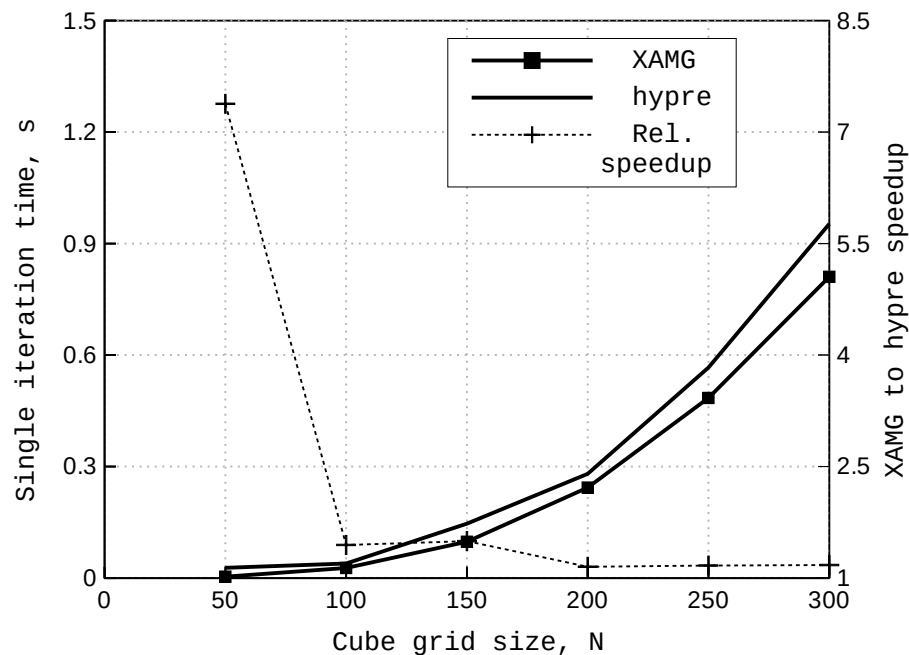
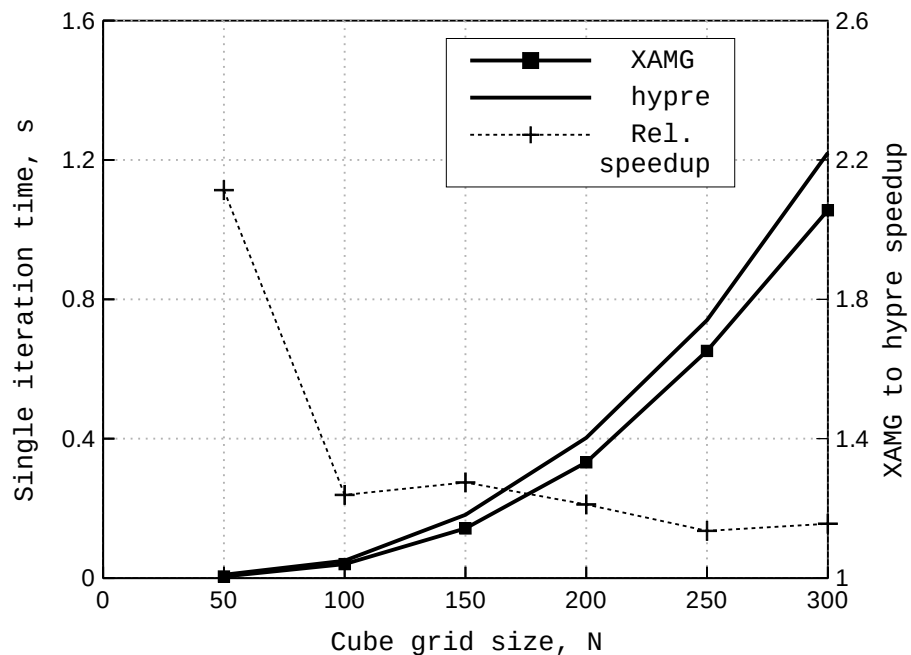
Содержание

План доклада:

1. При чем тут CFD?..
2. Общие принципы дизайна кода библиотеки
3. Математические методы
4. Особенности распараллеливания
- 5. Некоторые результаты тестирования**
6. Планы, проблемы...



Результаты: сравнение с hypre

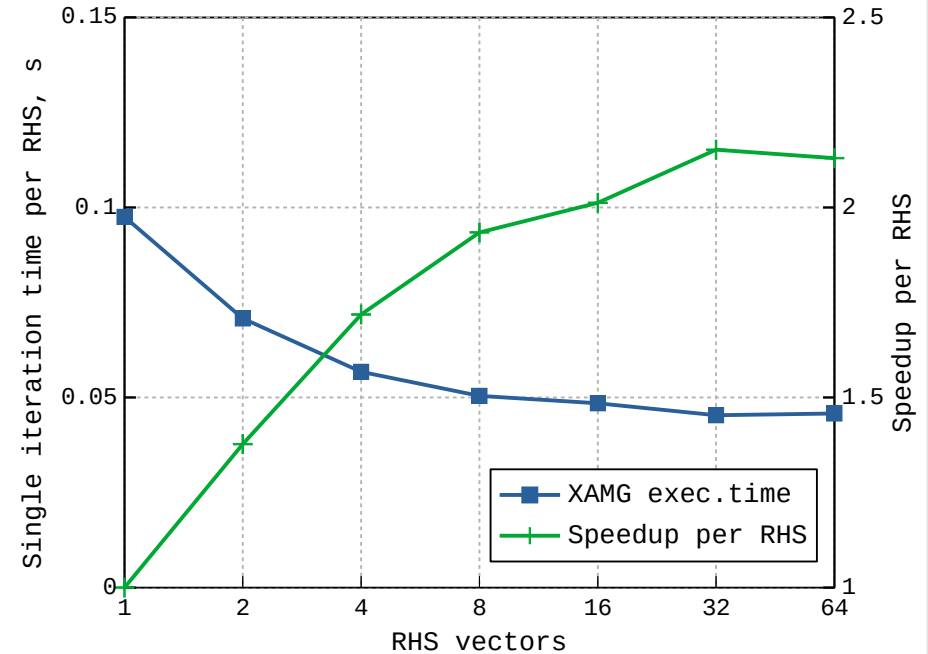
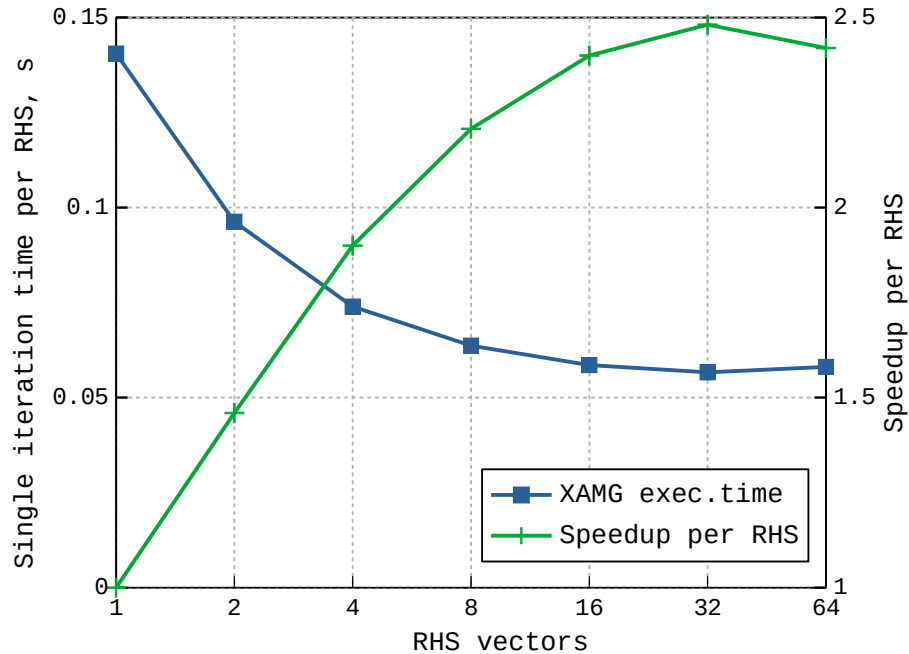


MPI, 1 узел (все ядра), $m=1$, зависимость от размера задачи (Лом-2 & НРС-4)

В среднем, на 20-30% быстрее, чем *hypre*



Результаты: зависимость от кол-ва правых частей

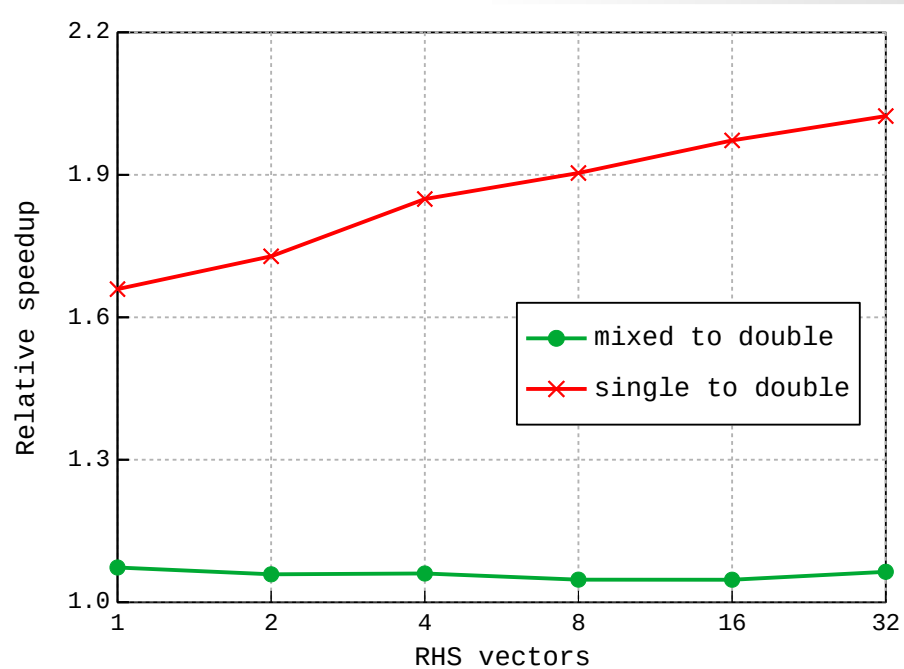
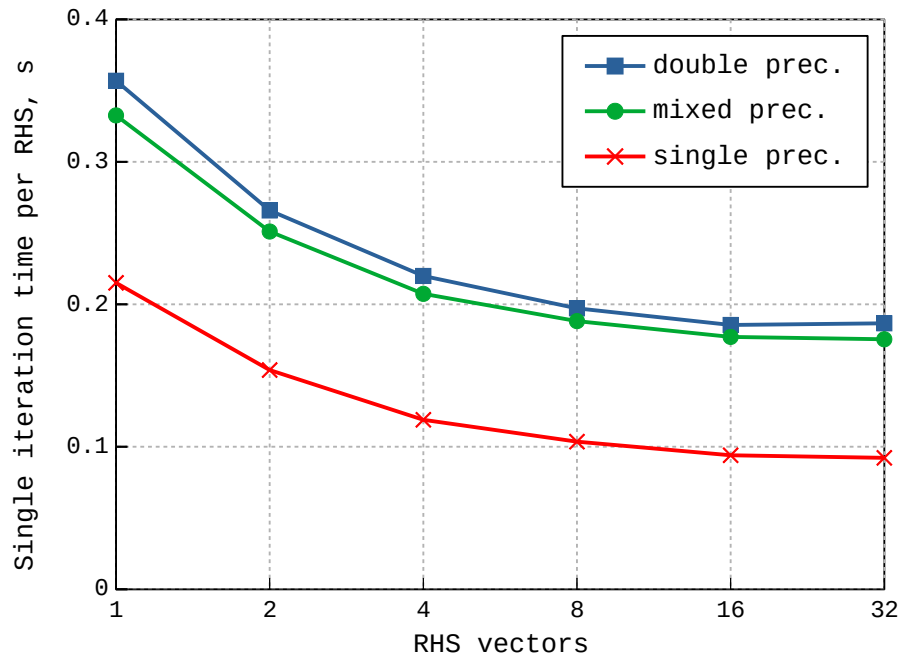


MPI, 1 узел (все ядра), $N=150^3$, влияние количества правых частей (Лом-2 & НРС-4)

Макс. выигрыш в пересчете на один вектор – для $m=32$, ускорение в 2.2-2.5 раза



Результаты: точность представления чисел с плавающей точкой

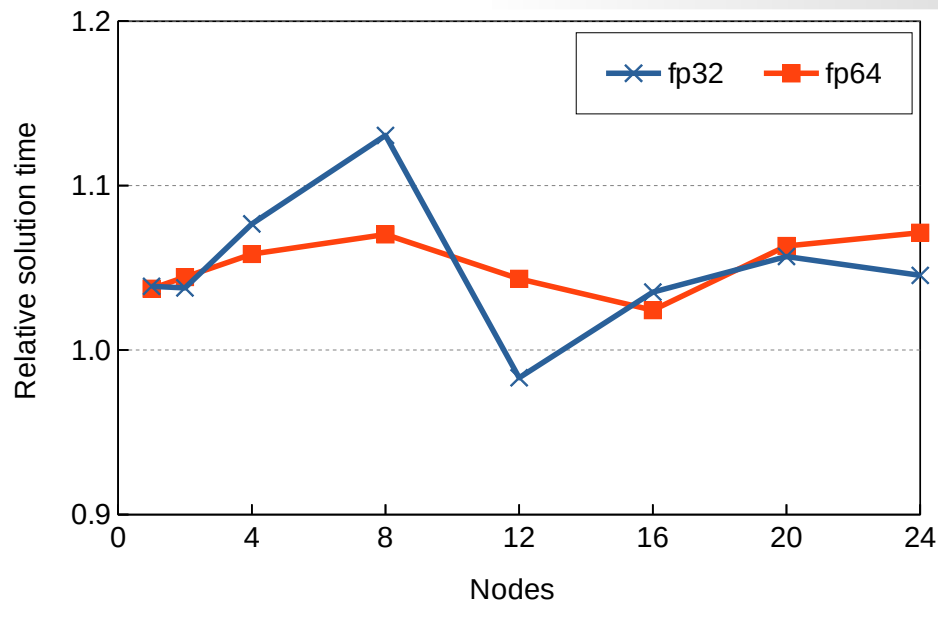
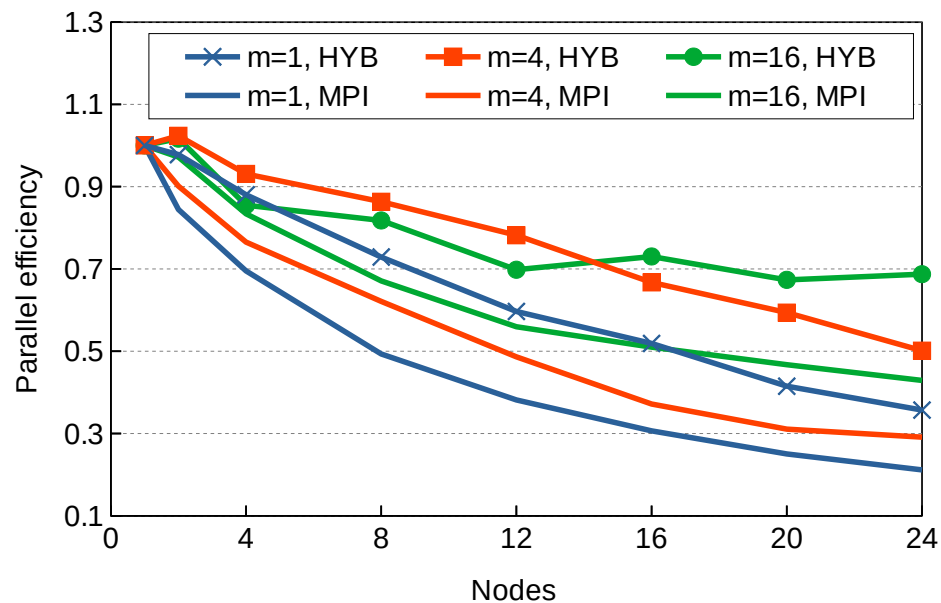


MPI, 1 узел (все ядра), $N=200^3$, fp64, fp64/32, fp32 (Лом-2)

Ускорение для одинарной точности растет с увеличением m , превосходя 2.



Результаты: роль гибридной модели

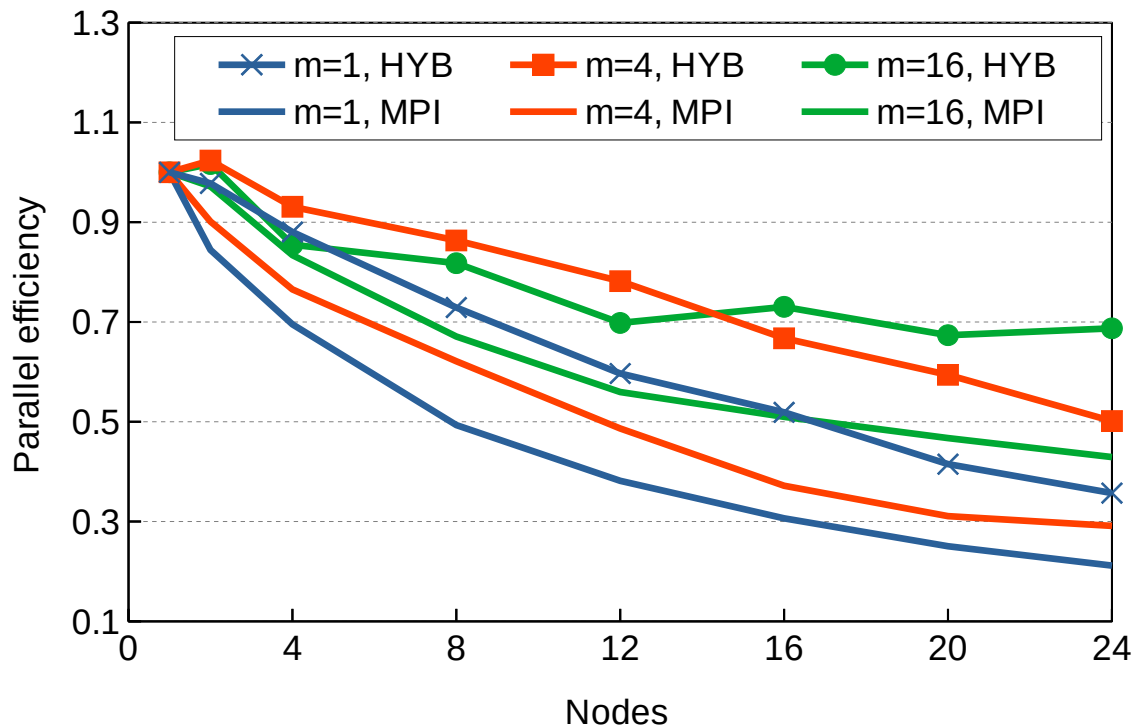


MPI и гибридная модель (2 numas:24 cores), $N=200^3$, fp64 (МСЦ)

Улучшение параллельной эффективности для гибридной модели до 2 раз



Результаты: роль гибридной модели

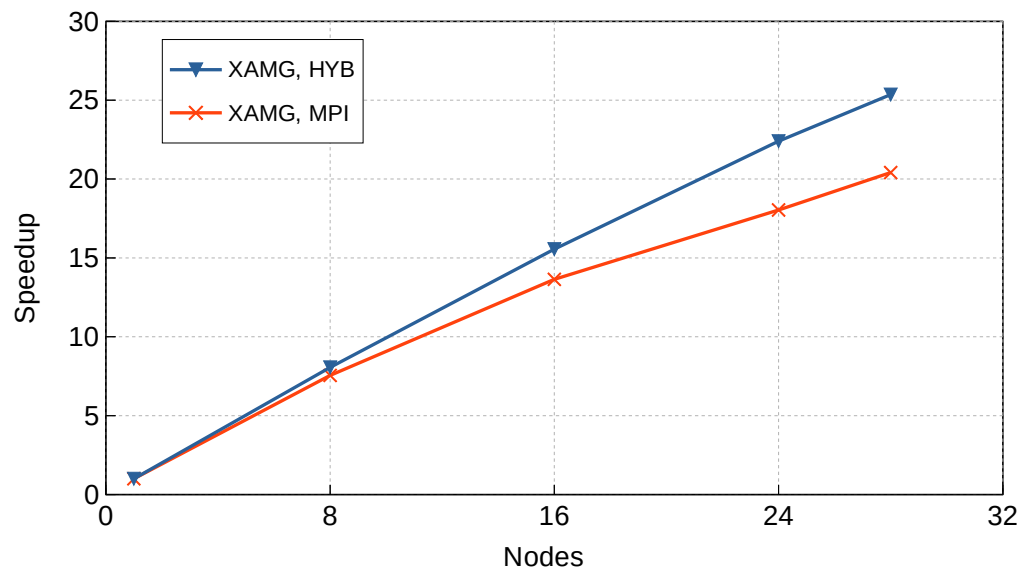
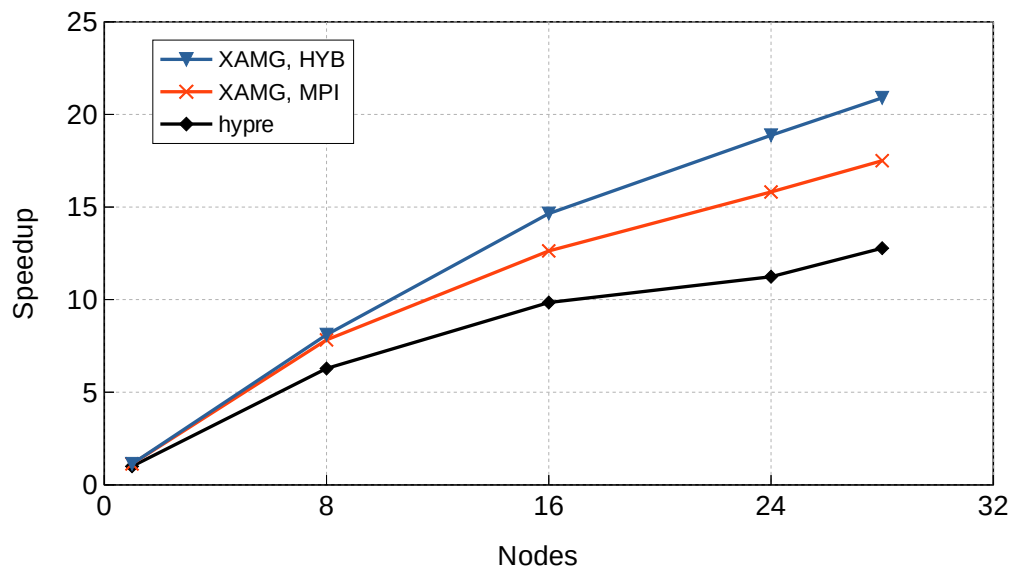


MPI и гибридная модель (2 numas:24 cores), $N=200^3$, fp64 (МСЦ)

Улучшение параллельной эффективности для гибридной модели до 1.5-2 раз



Результаты: задача обтекания куба в плоском канале



МРІ и гибридная модель (1 numa:12 cores), N=2.3М и 9.7М, fp64 (Лом-2)

Улучшение результатов *hypre* порядка 2 раз



Содержание

План доклада:

1. При чем тут CFD?..
2. Общие принципы дизайна кода библиотеки
3. Математические методы
4. Особенности распараллеливания
5. Некоторые результаты тестирования производительности
6. Планы, проблемы...



Планы

Дальнейшие планы развития библиотеки:

- * Реализация ветки кода для графических ускорителей
- * Локальные оптимизации
 - ↪ Форматы хранения данных
 - ↪ Разные варианты обменов данными между процессами
- * Автоматическая оптимизация параметров методов решения СЛАУ



Проблемы и задачи

Основные проблемы и насущные вопросы:

- * Вычислительные ресурсы...
- * Автоматизированное тестирование кода
- * Применение блочных итерационных методов
 - ↪ Возможно ли на практике ускорение сходимости для предобусловленных методов?



Доступно на gitlab

На условиях лицензии GPLv3
исходный код доступен по
ссылке:

<https://gitlab.com/xamg/xamg>

- скрипты для компиляции
- примеры...

XAMG > XAMG

XAMG @
Project ID: 22547337

2 Commits | 1 Branch | 0 Tags | 399 KB Files | 399 KB Storage

master xamg / +

Initial version of XAMG source code
Boris Krasnopolsky authored 12 minutes ago

README LICENSE Add CHANGELOG Add CONTRIBUTING Enable Auto DevOps Add Kubernetes cluster

Set up CI/CD

Name	Last commit	Last update
Make.Inc	Initial version of XAMG source code	12 minutes ago
ThirdParty	Initial version of XAMG source code	12 minutes ago
api/c	Initial version of XAMG source code	12 minutes ago
examples	Initial version of XAMG source code	12 minutes ago
src	Initial version of XAMG source code	12 minutes ago
tools	Initial version of XAMG source code	12 minutes ago
.gitignore	Initial version of XAMG source code	12 minutes ago
.gitmodules	Initial version of XAMG source code	12 minutes ago
LICENSE.GPL3	Initial version of XAMG source code	12 minutes ago

Выполнено в рамках гранта РФФ 18-71-10075



Спасибо за внимание!