

Проблемы адаптации приложений для
гетерогенных суперкомпьютеров на примере
задачи механики сплошных сред

Александр Монаков

amonakov@ispras.ru

Владимир Платонов

soh@ispras.ru

Арутюн Аветисян

arut@ispras.ru

Институт Системного Программирования

21 октября 2014 г.

Гетерогенные системы

- Нарращивание вычислительной мощности суперкомпьютеров: увеличение количества узлов/процессоров
- Повышение производительности процессоров затруднено:
 - Ограничения по тепловыделению
 - Приоритет на производительности однопоточного кода

Возможное решение: гибридная архитектура

- Дополнить центральный процессор акселератором
- Архитектура акселератора оптимизируется под массивно-параллельные задачи
- GPU: NVIDIA Tesla, AMD FirePro
- Intel MIC

- Специализированная массивно-параллельная архитектура
- 1-4 TFLOPS, 100-300 GB/s на одном устройстве
- Требуют задач с высоким параллелизмом

Преимущества

- 1 Производительность
- 2 Энергоэффективность (10x производительность при 2x энергопотреблении)
- 3 Доступность
 - В Top500: 62 системы с акселераторами, 44 NVIDIA
 - В персональных компьютерах

Крупный пакет для научных расчетов

- Задачи вычислительной гидродинамики (CFD)
- Открытый исходный код, активное сообщество
- Подготовка сеток
- Постпроцессинг и визуализация в ParaView
- Параллельные расчеты: MPI

Структура OpenFOAM

- 1 Программа-солвер для класса задач (pisoFoam, simpleFoam)
- 2 Модели расчетов (PISO, SIMPLE, ...)
- 3 Методы интерполяции, схемы дискретизации, ...
- 4 Методы решения линейных систем
 - Многосеточные методы
 - Метод сопряженных градиентов

Использование в OpenFOAM

50-80% времени счета: решение СЛАУ

- PCG: Метод сопряженных градиентов
- AMG: Многосеточный метод

Рассматриваем PCG

- Производительность ограничена:
 - Пропускной способностью памяти
 - Скоростью сети
- Сложности при переносе на GPU
 - Предобуславливание
 - Поддержка MPI
 - Минимизация накладных расходов

Преобуславливание на GPU

OpenFOAM: обычно используется DILU

- Подготовка/применение: решение треугольных систем
- Эффективный, но не обладает высоким параллелизмом
- Переупорядочивание может повысить параллелизм за счет снижения эффективности

Диагональное преобуславливание:

- Подготовка/применение: деление на диагональные коэффициенты
- Высокий параллелизм, низкая эффективность

Предобуславливание на GPU

Наш выбор для GPU: AINV

$$W^T AZ = D + e$$

- Подготовка: вычисление явного разложения приближенной обратной матрицы
- Применение: домножение на две разреженные матрицы
- Хорошо подходит для GPU
- Затраты на подготовку соизмеримы с решением СЛАУ

Асинхронное вычисление AINV

Предположим:

- 1 Что связность сетки не меняется
⇒ Портрет разреженной матрицы СЛАУ фиксирован
- 2 СЛАУ для последних шагов по времени имеют близкие коэффициенты
⇒ Можно переиспользовать предобуславливатели

Тогда:

- 1 Предобуславливатели вычисляются асинхронно в отдельном потоке
- 2 На GPU используется последний посчитанный предобуславливатель

Поддержка MPI

Схема параллельного счета в OpenFOAM:

- Разделение сетки (без перекрытия или “гало”)
- Межпроцессорные границы – частный случай граничных условий на сетке
- Процессоры обмениваются коэффициентами при домножении на матрицу системы
- Предобуславливание выполняется без MPI-коммуникаций

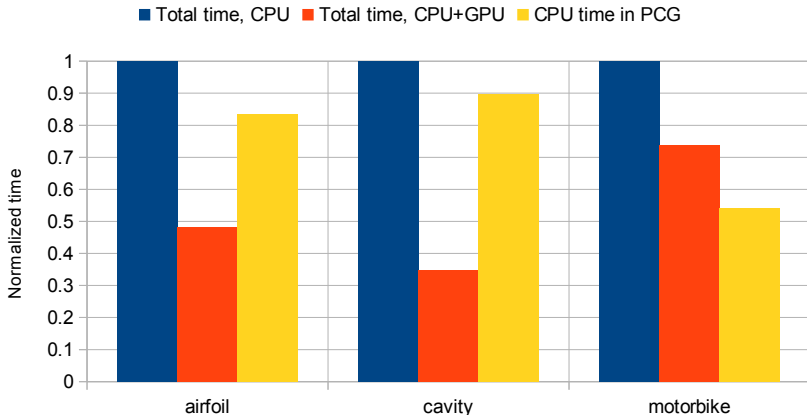
Поддержка на GPU:

- Сводится к поддержке произвольных граничных условий
- Параллельно с домножением на матрицу на GPU:
 - 1 Скопировать граничные коэф-ты в память CPU
 - 2 Вызвать методы обновления граничных условий
 - 3 Просуммировать обновленные коэф-ты на GPU

Снижение накладных расходов

- 1 Преобразование матрицы в GPU-формат только на первом шаге
На последующих шагах – обновлять ненулевые коэффициенты
- 2 Автоматическая настройка
 - Оптимизация умножения разреженных матриц на GPU
 - Разреженность предобуславливателя AINV
- 3 Минимизация CPU-GPU коммуникаций
 - Исключение лишних копирований
 - CUDA-aware MPI
 - Конвейеризованные варианты метода сопряженных градиентов

Производительность на 1 GPU

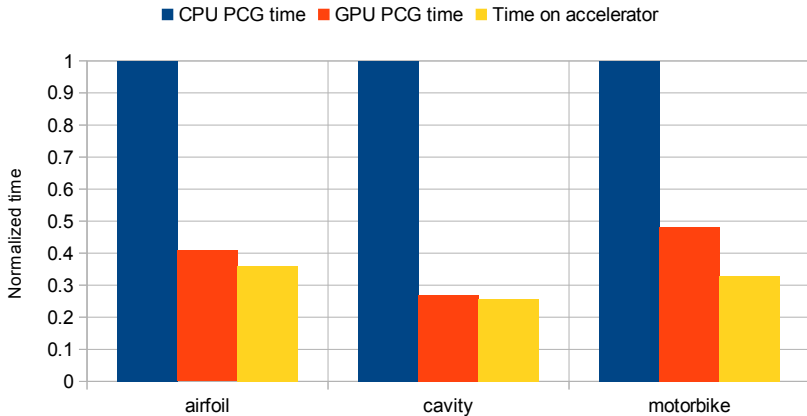


AIRFOIL, pisoFoam, 1.3M ячеек

CAVITY, icoFoam, 2M ячеек

MOTORBIKE, simpleFoam, 2.75M ячеек

CPU: 2x 6-core Intel Xeon, GPU: Tesla M2090



Case	$t_{\text{step}}^{\text{PCG}}$, ms	$t_{\text{startup}}^{\text{GPU}}$, ms	GPU Speedup	$N_{\text{iter}}^{\text{=}}$	N_{iter}^{2x}
AIRFOIL	14	150	2.8	17	75
CAVITY	30	130	3.9	6	18
MOTORBIKE	50	250	3.0	8	30

$t_{\text{step}}^{\text{PCG}}$: время на одну итерацию PCG

$t_{\text{overhead}}^{\text{GPU}}$: накладные расходы на копирование памяти CPU-GPU

$N_{\text{iter}}^{\text{=}}$: количество итераций, когда достигается паритет

N_{iter}^{2x} : количество итераций, когда GPU в 2 раза быстрее

Текущие работы

- Включение в релиз-версии FOAM-Extend
текущая версия доступна на Github¹
- Итерационные методы для асимметричных систем
- Решение блочных систем
- Дальнейшие оптимизации

¹<https://github.com/amonakov/openfoam-extend-foam-extend-3.1/tree/feature/ispmCudaSolvers/src/ispmCudaSolvers>

<http://www.unicluster.ru/>

Программа «Университетский кластер».
V Международная конференция
«Облачные вычисления. Образование.
Исследования. Разработка» 4-5 декабря

Chandrakant Patel,
HP Senior Fellow and Director Sustainable IT Ecosystem Laboratory,
Hewlett Packard Laboratories, Hewlett-Packard Company, (USA)



Thomas Kwan,
Director, Research Operations Yahoo! Labs, Yahoo! (CINA)



Евтушенко Юрий
Гаврилович академик РАН,
Директор ВЦ РАН



Иванников Виктор Петрович
академик РАН, Директор ИСП РАН



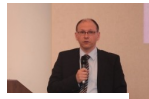
Michael A. Kozuch
Principal Engineer, Intel Labs
Pittsburgh



Dejan Milošić
Researcher, Intelligent
Infrastructure Lab, Hewlett-
Packard Company

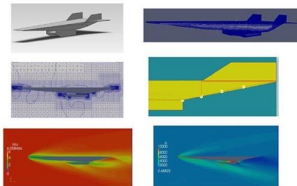


Dr. Henrik Rusche, WIKKI
Gesellschaft für numerische
Kontinuumsmechanik mbH



<https://unihub.ru/> Технологическая
платформа UniHUB.

<https://unicfd.ru/> Web-лаборатория UniCFD



Вопросы?

Спасибо!