

Разработка и реализация алгоритмов планирования вычислительной нагрузки на гетерогенный мультикомпьютер

Докладчик: Бе́ляев Н. А., маг. 2 курс НГТУ
Научный руководитель: Малышкин В. Э.,
д.т.н., проф.



Введение

- При расчете больших задач численного моделирования существенный прирост производительности может дать вовлечение в расчеты спец. вычислителей, таких как GPU
- Программирование взаимодействия между CPU и GPU, распределения вычислительной нагрузки между GPU и CPU является сложной задачей системного программирования, не связанной с предметной областью


Введение

- Частичная автоматизация конструирования параллельных программ, эффективно использующих доступные вычислительные ресурсы позволит абстрагировать разработчика параллельных программ от решения ряда задач системного параллельного программирования

Обзор существующих средств параллельного программирования с поддержкой GPU





- OpenCL
 - OpenACC
 - Charm++
 - DVMH
-
- На текущий момент от программиста требуется существенный объем знания  в области системного программирования для создания параллельных программ, эффективно утилизирующих  доступные вычислительные ресурсы

Цель работы

- Разработать алгоритм динамического распределения вычислительной нагрузки между GPU и CPU для системы LuNA, учитывающий  информационные зависимости программы с целью экономии коммуникаций между GPU и CPU
- Работа выполняется в рамках системы LuNA, т. к. система ориентирована на автоматизацию настройки программы на доступные ресурсы и может служить базой для исследования вопросов автоматизации



Задачи


- Разработать алгоритм распределения вычислительной нагрузки  для системы LuNA, учитывающий информационные зависимости между фрагментами  
- Разработать архитектуру модулей **распределения** вычислительной нагрузки
- **Реализовать** разработанные алгоритмы **распределения** вычислительной нагрузки в **виде**  модулей для системы LuNA
- Провести тестирование производительности системы со встроенными модулями

Система LuNA

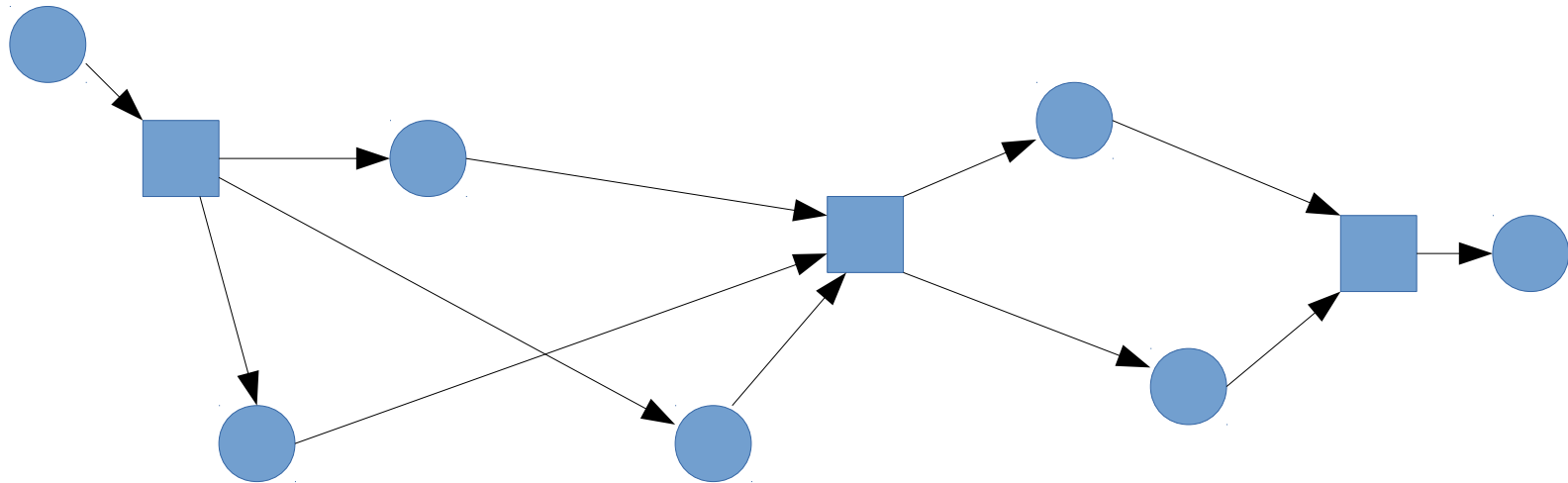
- Система конструирования параллельных алгоритмов LuNA разрабатывается в ИВМиМГ СО РАН и является системой параллельного программирования, ориентированной на решение задач численного моделирования на суперкомпьютерах



Параллельная программа в системе LuNA

- Параллельная программа в системе представляется в виде множества фрагментов данных (ФД, агрегированные переменные единственного присваивания) и множества фрагментов вычислений (ФВ, агрегированные операции единственного срабатывания)
- ФВ  потребляя значения входных ФД, вычисляют на их основе значения выходных ФД

Параллельная программа в системе LuNA



- ФД



- ФВ



- Информационная зависимость
между фрагментами

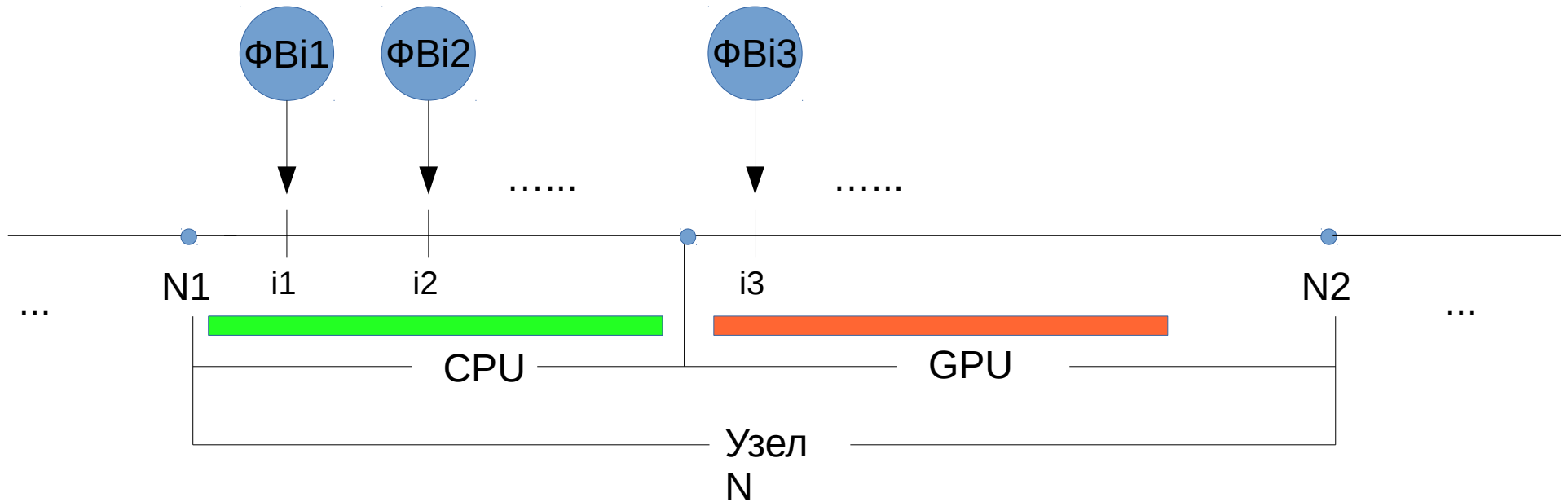
Задача распределения вычислительной нагрузки в системе LuNA

- Для каждого ФВ перед его исполнением принимается решение о выборе устройства для запуска ФВ (CPU или GPU)

Наивный алгоритм

- ФВ назначаются для исполнения последовательно на GPU и CPU в порядке их поступления на исполнение в системе

Схема предлагаемого алгоритма распределения вычислительной нагрузки MD-RoV

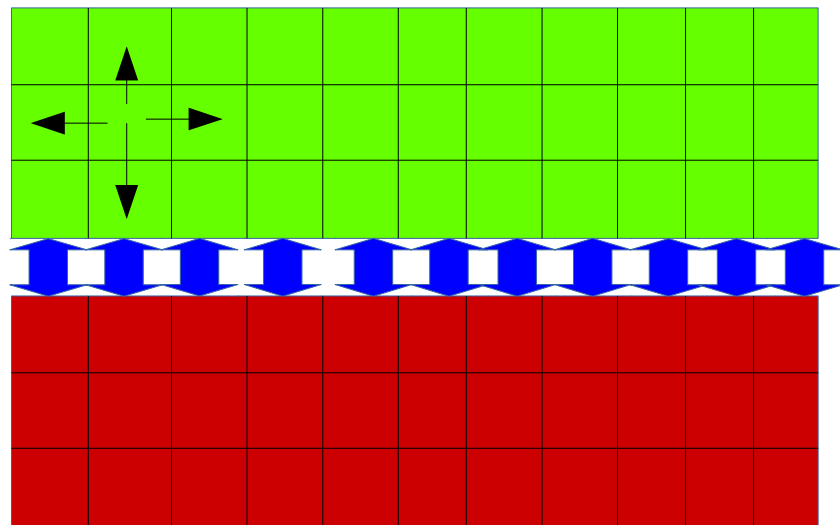


Автоматический выбор фрагментации

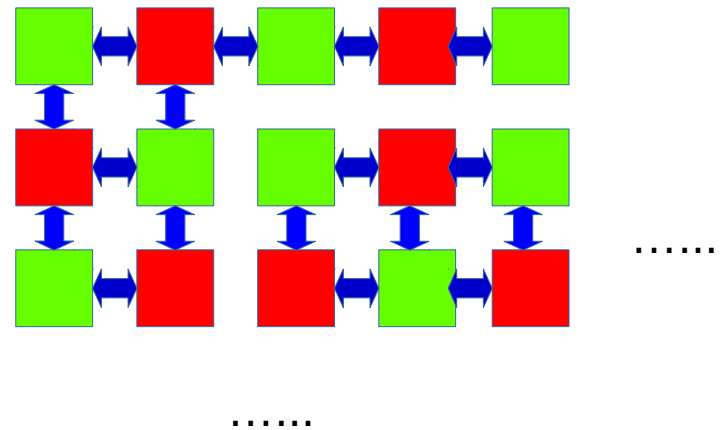





Преимущества алгоритма MD-RoV

Алгоритм RoV




Наивный алгоритм



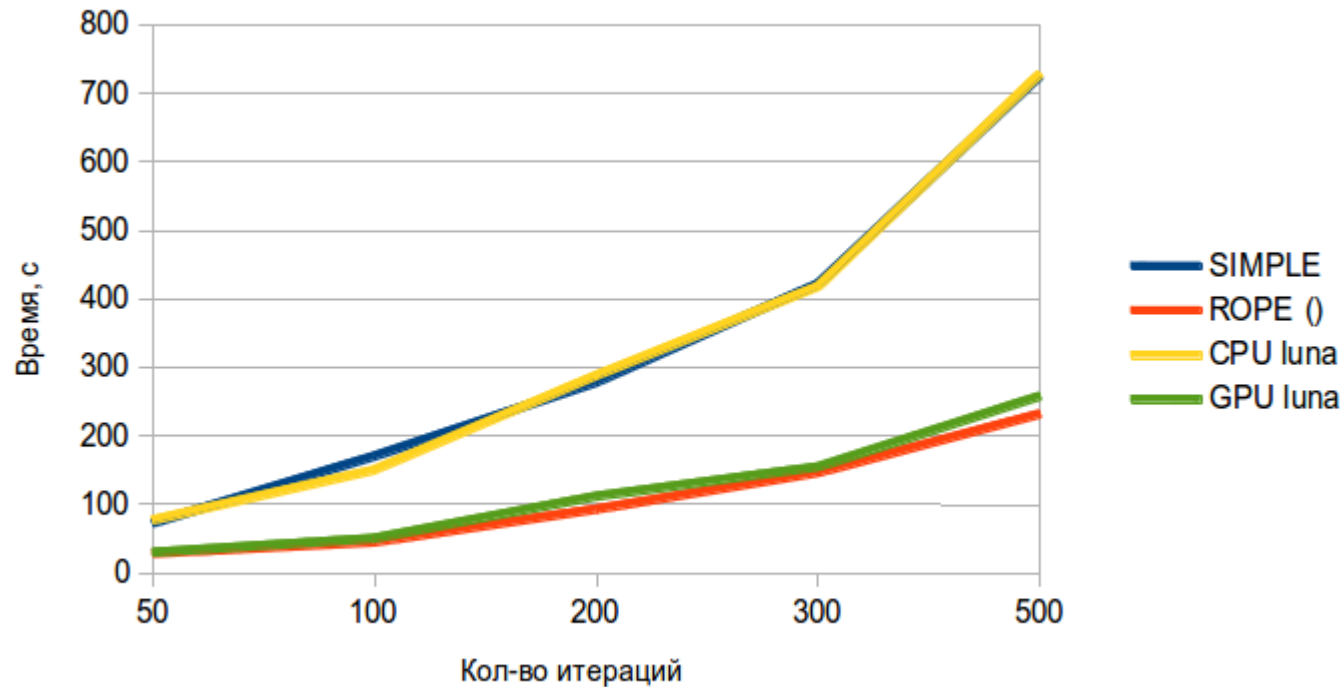
-  ФВ, выполняющийся на GPU
-  ФВ, выполняющийся на CPU
-  Передача данных между GPU и CPU

Тестирование

- Цель тестирования: оценить выигрыш в производительности при использовании алгоритма RoV 
- Тестовая задача: параллельное решение уравнения Пуассона методом Зейделя в 3D области с одномерной фрагментацией
- Выбор фрагментации : ручная


Тестирование

Размер задачи: 50 фрагментов x 129x129x129
float

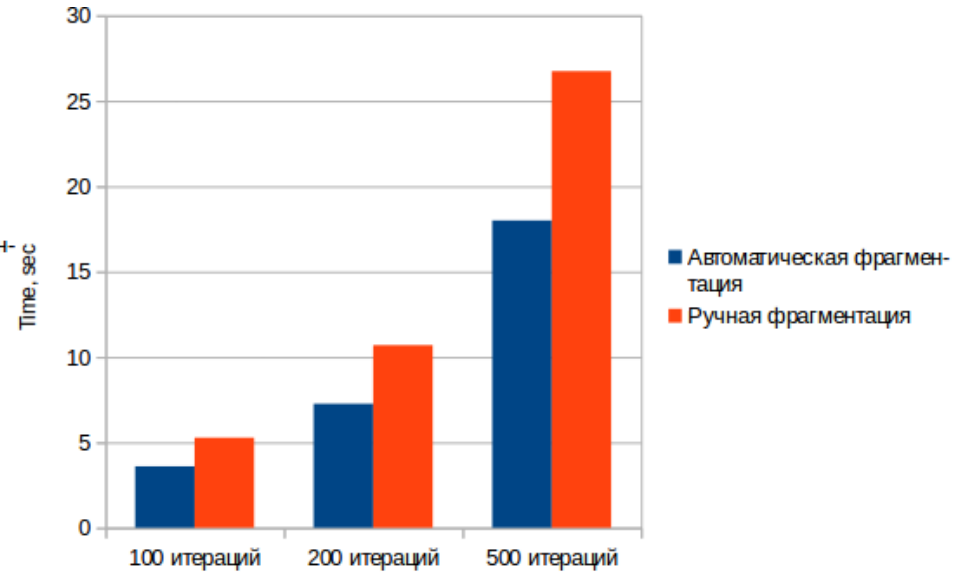
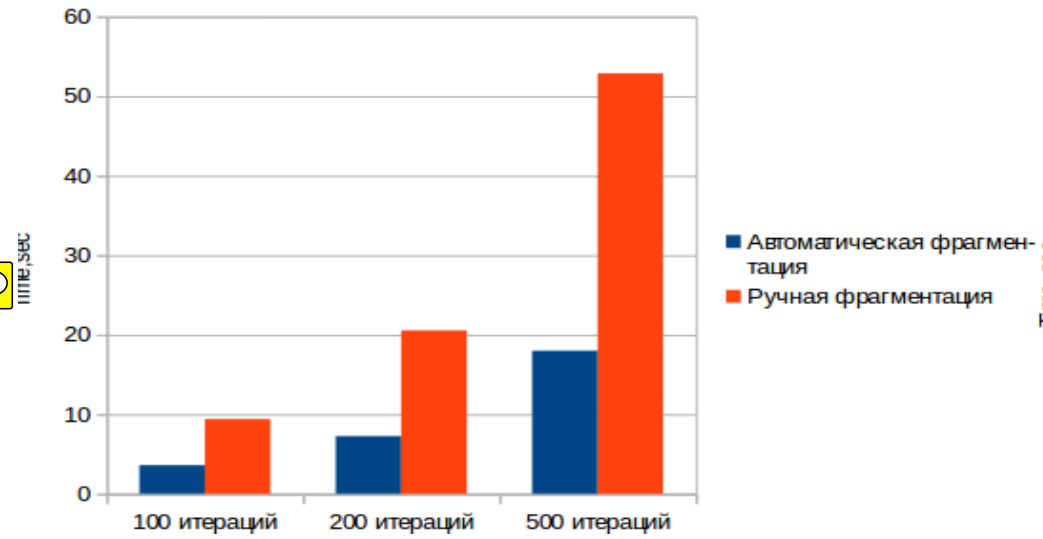


Оборудование: Intel Xeon X5660 2.8GHz, NVIDIA GT200GL [Quadro FX 4800]

Тестирование

- Выигрыш от применения автоматического выбора фрагментации
- Задача: решение уравнения Пуассона в 2Д методом Зейделя 
- Размер 100 фрагментов x 800x800 (double)
- CPU: Core i7 3820 @ 3.6 GHz 4 CORES
- GPU: Nvidia GeForce GTX 650 32x32 CUDA threads

Тестирование



Заключение

- Разработан алгоритм распределения вычислительной нагрузки на основе алгоритма RoV для системы LuNA
- Разработанные модули были реализованы в виде модулей для системы LuNA
- Проведенное тестирование показало прирост производительности системы LuNA при применении алгоритма MD-RoV по сравнению с наивным алгоритмом и реализациями фрагментированной программы, использующими только GPU или только CPU

Планы

- Разработать алгоритм автоматического выбора параметров алгоритма MD-RoB
- Разработать модули для поддержки FPGA в системе LuNA