

Распределение вычислительной нагрузки между GPU и CPU в системе LuNA

Докладчик: Беляев Н. А.,
ИВМиМГ СО РАН

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 знаний в области системного программирования для создания параллельных программ, эффективно утилизирующих доступные вычислительные ресурсы

Примечание

- 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

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

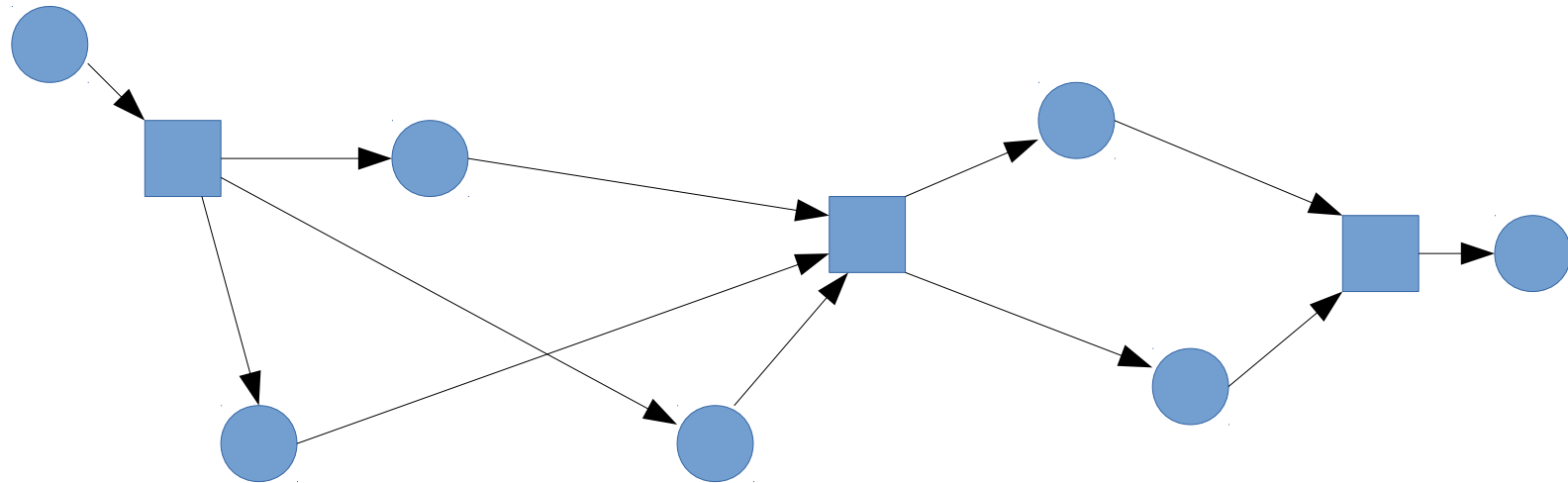
Задачи

- Разработать алгоритм распределения вычислительной нагрузки между GPU и CPU для системы LuNA, учитывающий информационные зависимости в программе с целью экономии времени на коммуникациях между GPU и CPU
- 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 программ LuNA разрабатывается в ИВМиМГ СО РАН и является системой параллельного программирования, ориентированной на решение задач численного моделирования на суперкомпьютерах

Параллельная программа в системе LuNA



- ФД



- ФВ



- Информационная зависимость
между фрагментами

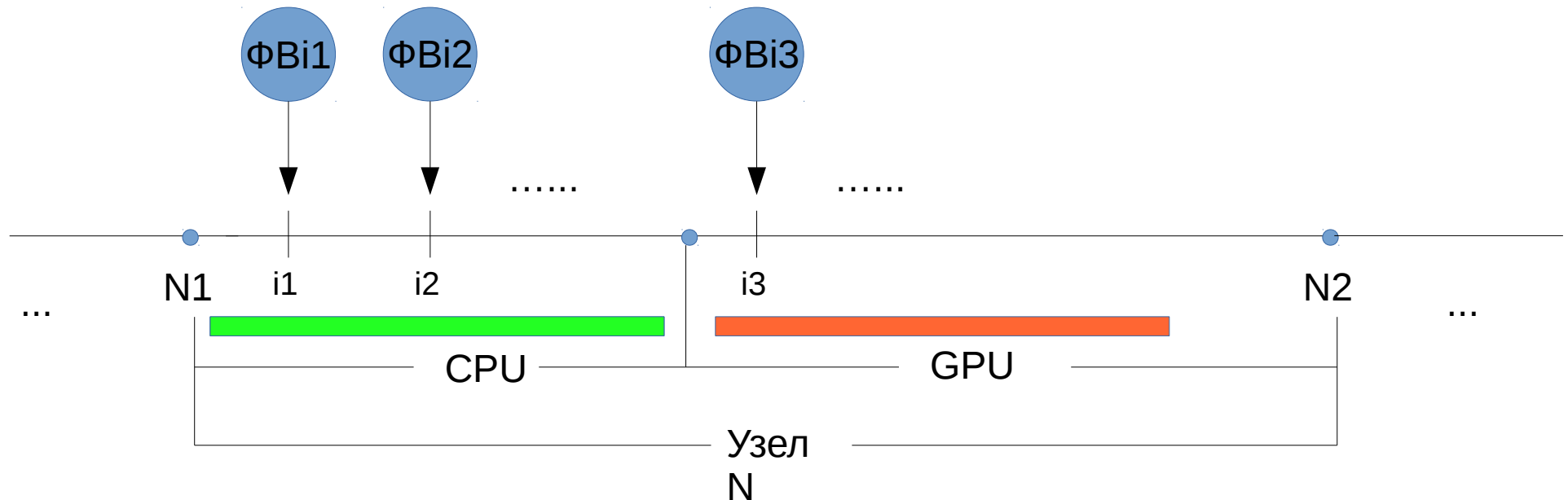
Задача распределения вычислительной нагрузки в системе LuNA

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

Наивный алгоритм

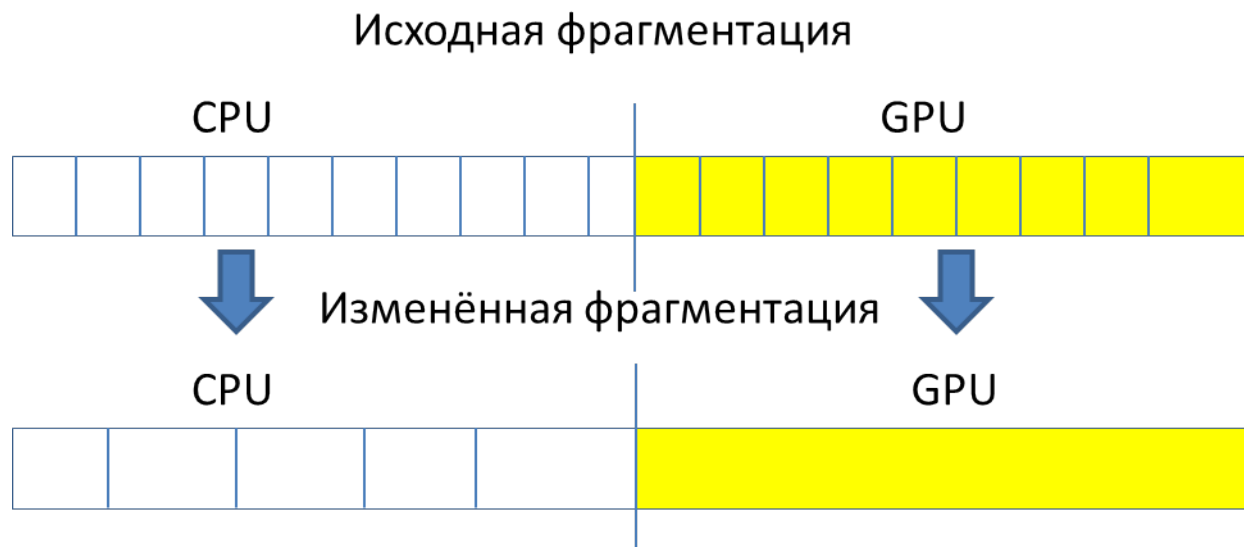
- ФВ назначаются для исполнения последовательно на GPU и CPU в порядке их поступления на исполнение в системе

Схема предлагаемого алгоритма распределения вычислительной нагрузки MD-RoB (Multiple Devices Rope of Beads)



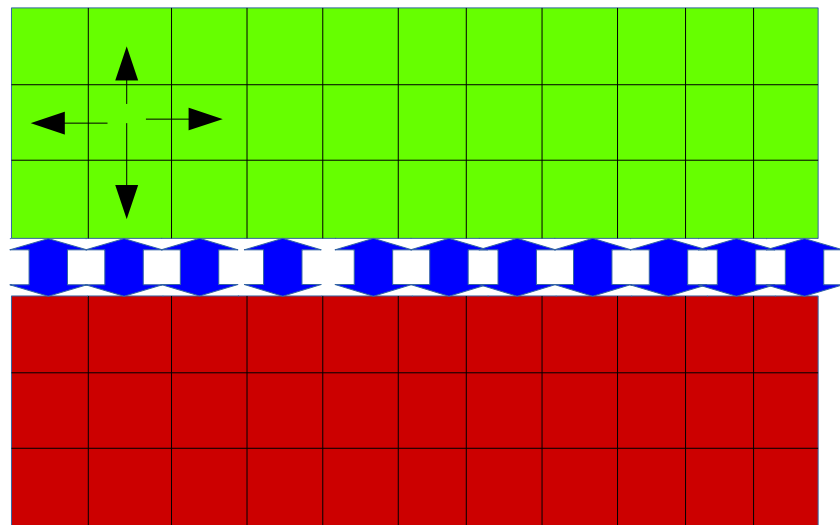
*Алгоритм основан на алгоритме RoB : Victor Malyshkin, Vladislav Perepelkin, and Georgy Schukin Distributed Algorithm of Data Allocation in the Fragmented Programming System LuNA // Proceedings of the 13th International Conference on Parallel Computing Technologies (PaCT-2015), LNCS, Vol. 9251.2015. P. 80-85.

Автоматический выбор фрагментации

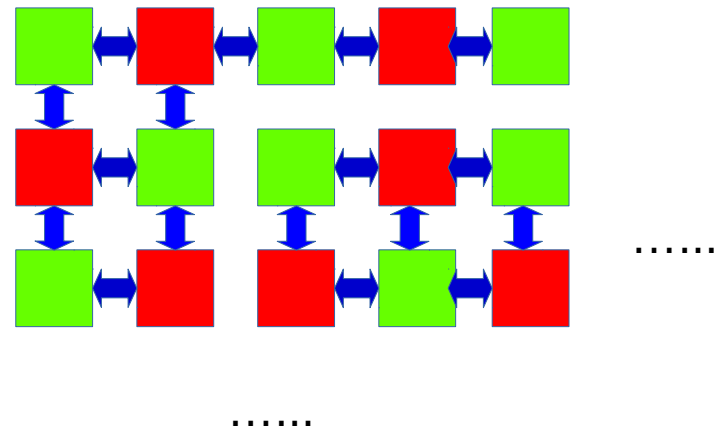





Преимущества алгоритма MD-RoV

Алгоритм RoV



Наивный алгоритм



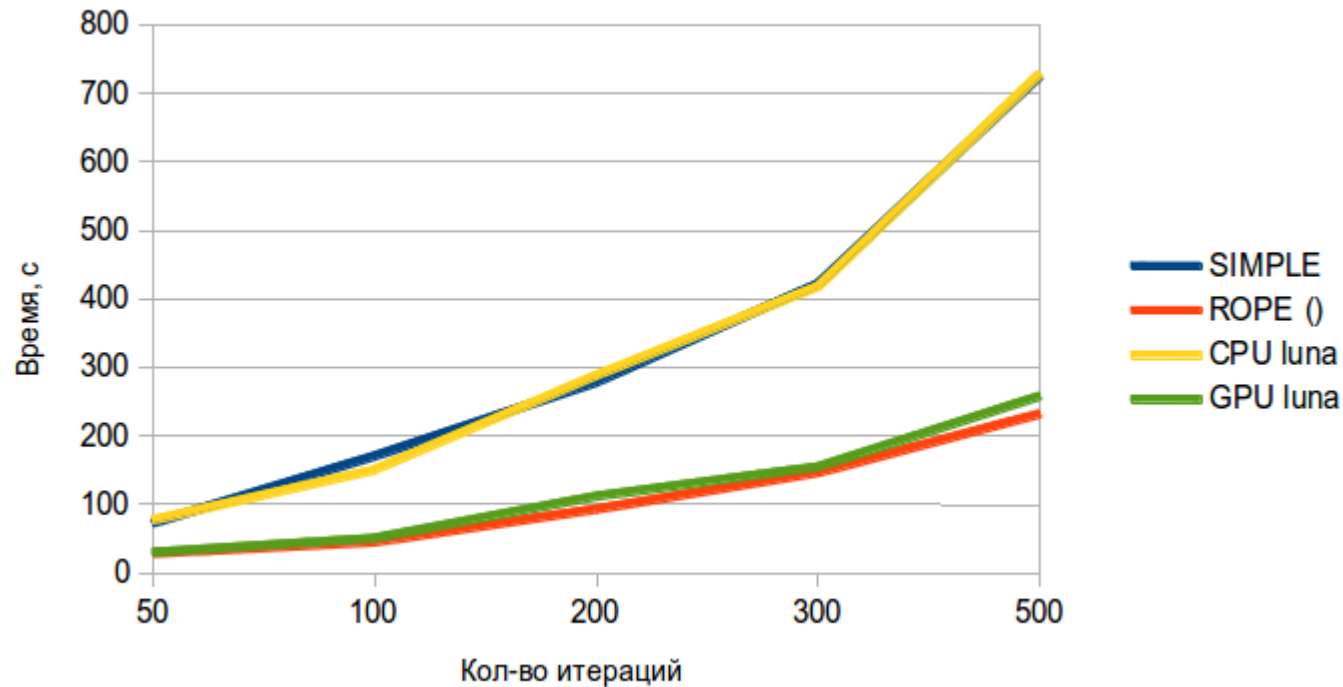
-  ФВ, выполняющийся на GPU
-  ФВ, выполняющийся на CPU
-  Передача данных между GPU и CPU

Тестирование

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

Тестирование

Размер задачи: 50 фрагментов x 129x129x129 float

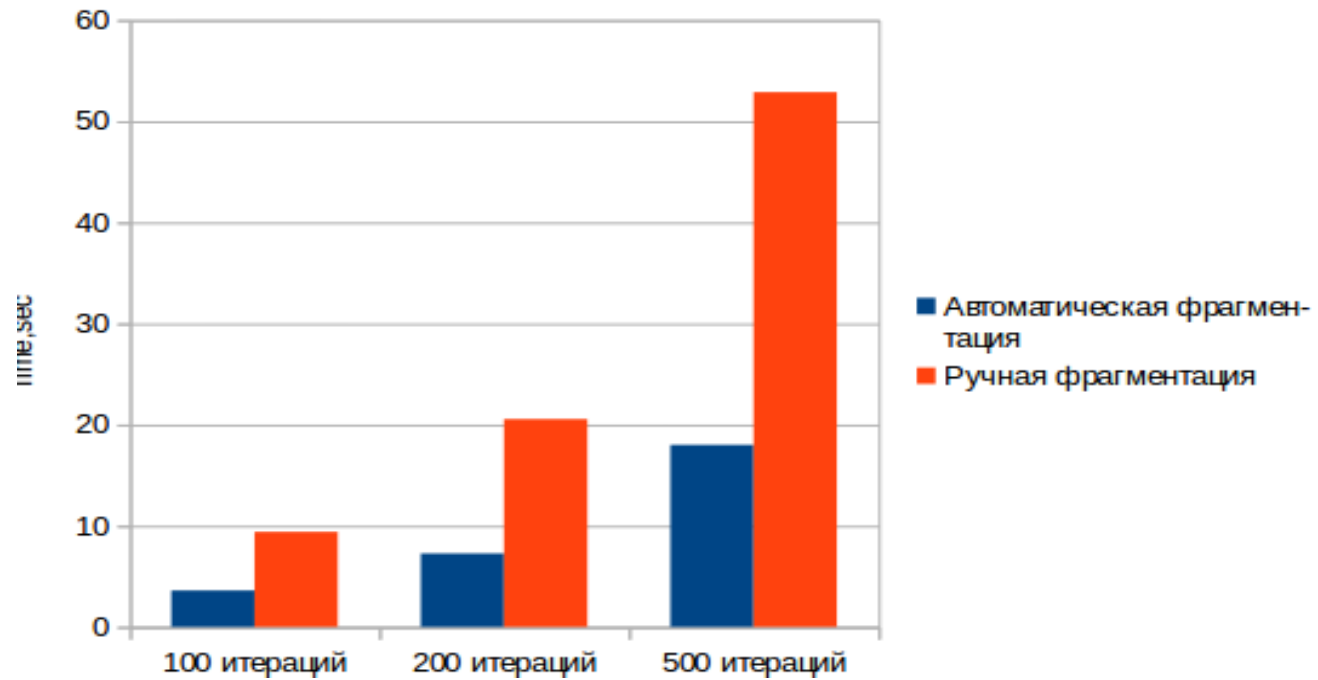


Оборудование: Intel Xeon X5660 2.8GHz, NVIDIA GT200GL [Quadro FX 4800]

Тестирование

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

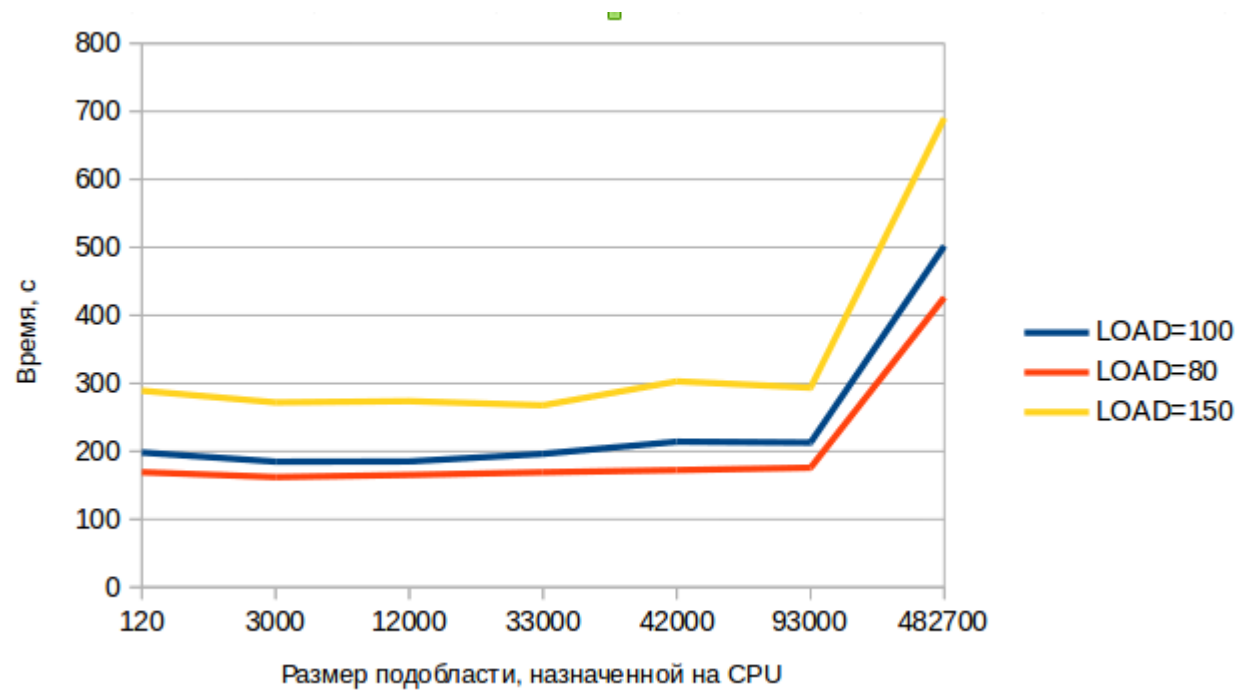
Тестирование



Тестирование

- Тест в распределенной памяти, 7 узлов
- Размер 483000x200 double
- Автоматический выбор фрагментации
- CPU: 2x Xeon X5670 2.93GHz
- GPU: 3x Nvidia TESLA M 2090 512 CUDA threads

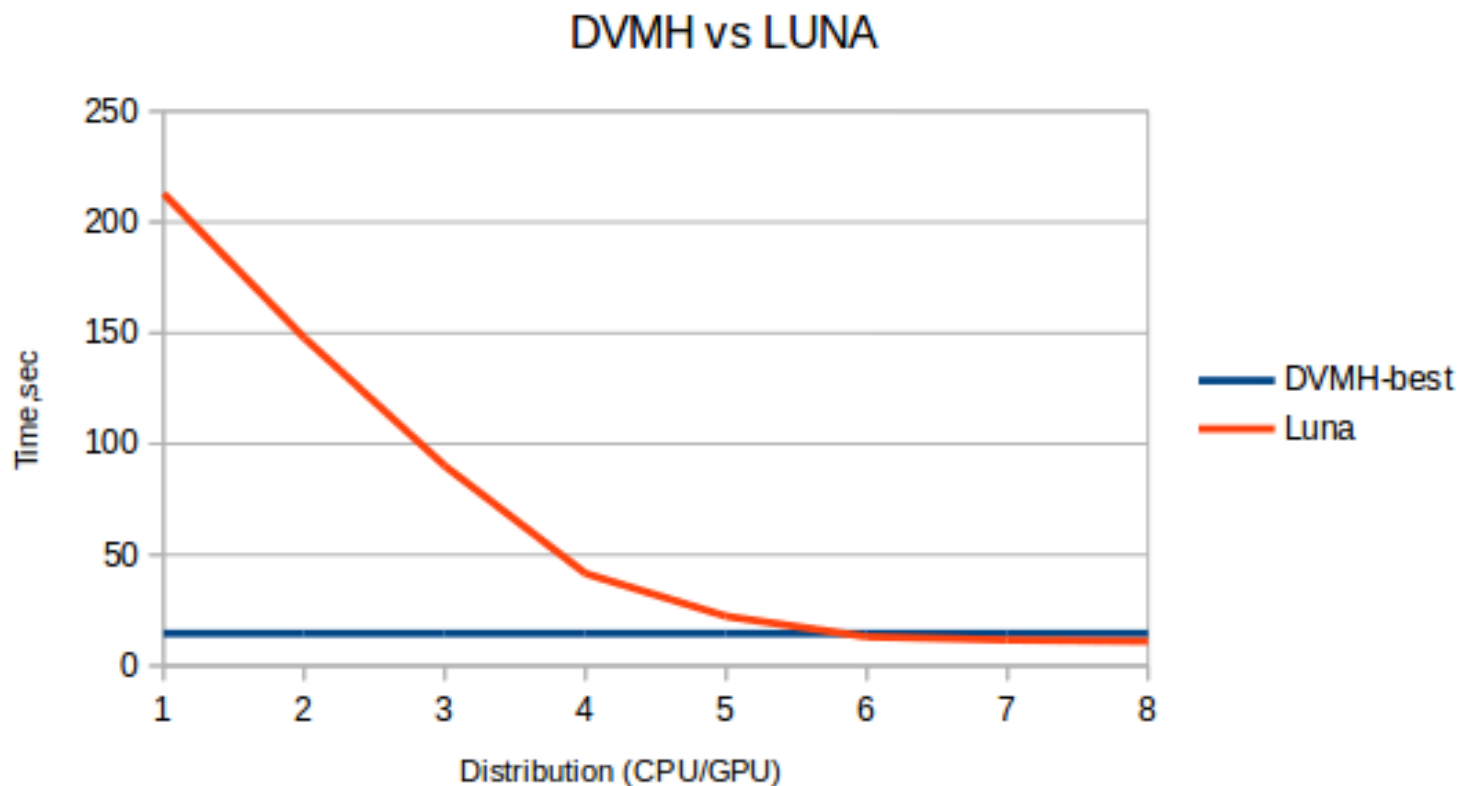
Тестирование



Тестирование

- Сравнение производительности системы LuNA и DVMH
- Размер расчетной области 40 фрагментов размера 2000x2000 элементов типа double
- Автоматический выбор фрагментации
- CPU: Intel Core i7 3820 @ 3.60GHz и GPU
- GPU: NVIDIA GeForce GTX 650

Сравнение производительности системы LuNA и DVMH



Заключение

- Разработан алгоритм распределения вычислительной нагрузки на основе алгоритма MD-RoV для системы LuNA
- Разработанные алгоритмы 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