Разработка и реализация алгоритмов эффективного исполнения фрагментированных программ на гетерогенном мультикомпьютере

Докладчик: Беляев Н. А., аспирант 1 курса ИВМиМГ СО РАН
Научный руководитель: д. т. н., проф., Малышкин В. Э.

Введение

- Решение задач численного моделирования с использованием суперкомпьютеров требует разработки параллельных программ
- Разработка параллельных программ для решения задач численного моделирования зачастую требует от разработчика навыков системного параллельного программирования

Введение

- Системное параллельное программирование не является частью предметной области
- Абстрагирование разработчика параллельной программы для решения задач численного моделирования от решения задач системного программирования позволит сократит время разработки параллельных программ

Требования к системам параллельного программирования

- Абстрагирование разработчика параллельной программы от зада системного параллельного программирования
- Обеспечение уровня производительности параллельных сравнимого с уровнем, достигаемым при ручной реализации программ

Существующие средства и системы ПП

- MPI
- OpenMP
- DVM-H, sapfor
- Charm++
- OpenCL

Система LuNA

Общие сведения

- Система LuNA – это система автоматического конструирования параллельных программ, ориентированная на решение задач численного моделирования на суперкомпьютерах

Параллельная программа в системе LuNA

- ПП в системе LuNA представляется я виде множества фрагментов вычисления (ФВ) и фрагментов данных (ФД)
- Также ПП в системе LuNA может быт представлена в виде двудольного ориентированного графа, вершинами которого являются ФВ и ФД, а дуги обозначают информационные зависимости

Термины и определения

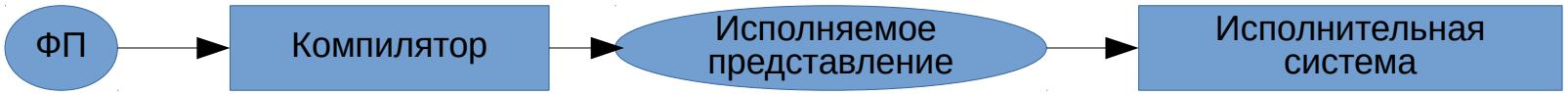
- Управление (на множестве ФВ) - отношение частичного порядка (на множестве ФВ), накладывающее ограничения на порядок исполнения ФВ

Исполнение фрагментированной программы

- Исполнение ФВ в системе LuNA происходит по т. н. Базовому алгоритму, ФВ исполняется по готовности значений всех своих входных ФД
- В языке LuNA присутствуют ФВ двух видов:
  - Атомарные ФВ, представляющие собой вызовы процедур языка Си
  - Структурированные ФВ (if, for, while, sub) – представляют собой ФВ, исполнение которых заключается в поступлении на исполнение некоторого множества ФВ, называемого телом структурированного ФВ. Исполнение структурированного ФВ в дальнейшем условно будем называть “раскруткой” ФВ

Архитектура системы LuNA

- Система состоит из компилятора языка LuNA и исполнительной системы



Недостатки (проблемы) системы LuNA

- Неконтролируемая “раскрутка” структурированных ФВ во время исполнения ФП
- Неэффективные алгоритмы распределения фрагментов по узлам мультикомпьютера.
- Проблема «сборки мусора»
- Использование универсальных алгоритмов динамической балансировки вычислительной нагрузки на узлы мультикомпьютера
- Решения о выборе управления на подмножествах ФВ принимается динамически, что снижает производительность системы

Система LuNA-2

LuNA-2

- Система LuNA-2 должна автоматически конструировать параллельные программы, обладающие свойством настраиваться на конкретный вычислитель (в т. ч., свойством обеспечения при необходимости динамической балансировки вычислительной нагрузки) для заданного класса численных алгоритмов
- Производительность сконструированных программ не должна быть хуже таковой у программ, разработанных “вручную” более чем в 2 раза

Цель работы

- Разработка системных алгоритмов эффективного исполнения фрагментированных программ на мультикомпьютере для заданного класса задач численного моделирования и реализация алгоритмов в виде средства параллельного программирования рамках системы LuNA

Задачи

Разработать алгоритмы распределения фрагментов данных и фрагментов вычислений на узлы мультикомпьютера

- Разработать алгоритмы анализа ФА для выбора походящего алгоритма распределения фрагментов на узлы мультикомпьютера
- Разработать алгоритм анализа результатов измерения производительности системы во время предыдущих запусков фрагментированной программы (ФП) (т. н. профиля исполнения ФП)
- Разработать алгоритмы динамического перераспределения фрагментов на узлы мультикомпьютера (алгоритмы динамической балансировки вычислительной нагрузки)

Задачи

- Разработать алгоритмы выбора алгоритма динамической балансировки для множеств фрагментов, при исполнении которых наблюдается дисбаланс вычислительной нагрузки
- Разработка алгоритмов своевременного освобождения памяти, занимаемой значениями ФД, уже потребленными всеми ФВ, для которых данное значение было входным (алгоритмы «сборки мусора»)
- Разработка алгоритмов выбора управления, согласно которому будет осуществлено исполнение фрагментов на узлах мультикомпьютера.
- При выборе управление алгоритмы должны по возможности сокращать общее время исполнения ФП и потребление ресурсов мультикомпьютера во время исполнения. В частности, необходимо, по возможности, выбрать управление, позволяющее максимально быстро освобождать память, занимаемую значениями ФД

Детали реализации системы LuNA-2

Общие сведения

- Система LuNA-2 на текущий момент представляет собой интеллектуальный компилятор
- Компилятор выделяет из ФА множество фрагментов, для которых возможна генерация статического управления и распределения ресурсов и осуществляет генерацию в случае, если это возможно

Общие сведения

- Подмножества фрагментов, для которых генерация статического управления и распределения ресурсов, исполняются системой LuNA
- Для облегчения анализа ФА и повышения уровня программирования язык LuNA был дополнен

Изменения языка LuNA

- Ключевое слово reduction, пример:<reduce max>(D,diff, real, cf_reduce(%in, %out));
- Ключевое слово borders_exchange, пример:<borders_exchange 1>(Fi(it), B, read(%in, %1, %2), write(%out, %1, %2, %FRAG_IDX));
- Сущность «массив ФД», пример: DFArray A[n][n];
- Явное выделение итерационного процесса над множеством массивов ФД в описании ФА: while (it < 500), it = 0 .. out iter : <Fi(it) --> Fi(it+1)>{...}

Текущие ограничения, накладываемые на ФА

- Размеры всех массивов ФД имеют одинаковый размер
- Размерность всех массивов ФД
- Доступ к элементам массивов ФД имеет вид A[df1][df2]...[dfN]
-

Требования к архитектуре компилятора LuNA-2

- Модульность
- Универсальность внутреннего представления ФА
- Универсальность модулей (каждый модуль должен содержать реализацию преобразования над универсальным внутренним представлением и может быть применён к нему на любой стадии компиляции)

Требования к внутреннему представлению

- Универсальность
- Возможность представления информации о семантической структуре ФА
- Возможность представления информации о подмножествах ФВ, для которых возможно параллельное исполнение и требуется распределение на узлы мультикомпьютера

Внутреннее представление P2CR (основные определения)

- P2CR (Phase 2 Common Representation)- основная структура данных фазы 2
- P2CR - Ориентированный граф (возможны циклы)
- Область — совокупность непустого множества массивов ФД, имеющих одинаковую размерность и одинаковые размеры по каждому измерению и множества ФД, значения которых задают размерности массивов
- Блок — множество ФВ
- Вершины соответствуют ФД, т. н. областям и т. н. блокам, а ребра — информационным зависимостям между блоками и областями
- Дуга от области (блока) к блоку (области) означает, что ФВ блока потребляют (вырабатывают) значения ФД или значения элементов массивов ФД области согласно некоторому множеству выражений языка LuNA (эти выражения являются «меткой» каждой дуги графа)
- Дуга от ФД (блока) к блоку (ФД) имеет аналогичный смысл

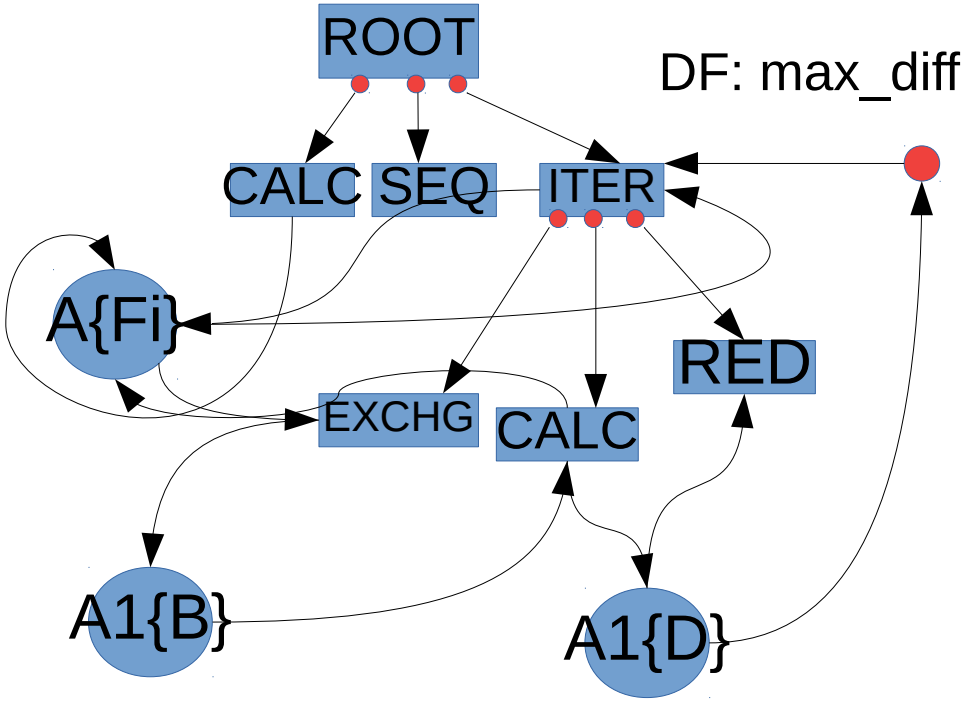
Внутреннее представление P2CR (основные определения)

- Дуга от блока к блоку задает либо прямое управление между блоками, либо указывает на то, что ФВ блока поступают на исполнение вследствие исполнения структурированного ФВ, содержащегося в блоке-начальной вершине дуги, в зависимости от метки, присвоенной дуге

Внутреннее представление P2CR

- Блоки имеют тип аналогично типу ФВ блока:
  - ITER — блок, содержащий ФВ, задающий итерационный процесс на расчётной области (содержит ФВ типа for или while)
  - CALC — блок, содержащий множество ФВ, реализующих непосредственно вычисления на расчётной области
  - SEQ — блок, содержащий ФВ, которые необходимо выполнить последовательно на одном или нескольких из узлах мультикомпьютере
  - RED — блок, содержащий ФВ, реализующие редукцию на массиве ФД
  - EXCHG — блок, содержащий ФВ, реализующие обмен теневыми гранями между ФД

Внутреннее представление P2CR




```
sub main()
{
  df iter, max_diff(2);
  DFArray Fi[500];
  for i=0..499 {
    init_fi(Fi(0)[i], i);
  }
  while (it < 500 ), it = 0 .. out iter : <Fi(it) --> Fi(it+1)>
  {
    df diff;
    DFArray B[500], D[500];
    <borders_exchange 1>(Fi(it), B, read(%in, %1, %
          write(%out, %1, %2, %FRAG_IDX));
    assign(max_diff(it+1), diff);
    for i = 0 .. 499 {
      cf clc[i] : poi(B[i], i, Fi(it+1)[i], D[i]);
    }
    <reduce max>(D,diff, real, cf_reduce(%in,  %out)
  }
init_max(max_diff(0));
}
```

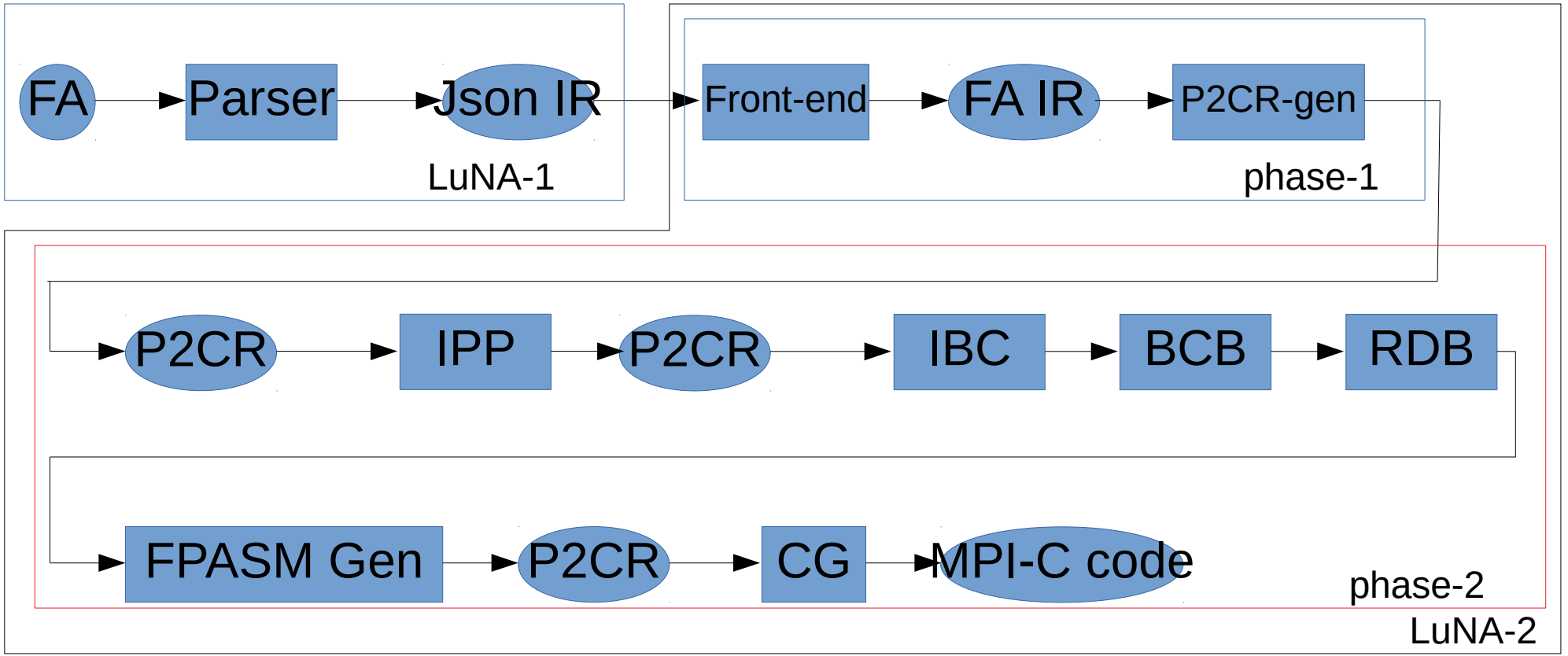
Внутреннее представление FPASM

- FPASM (Fragmented Program Assembler) — представляет собой множество команд, на котором задано отношение частичного порядка (управление), определяющее порядок исполнения команд

Внутреннее представление FPASM

- Краткий список команд FPASM:
  - Исполнение ФВ
  - Исполнение множества ФВ
  - Редукция
  - Обмен теневых граней

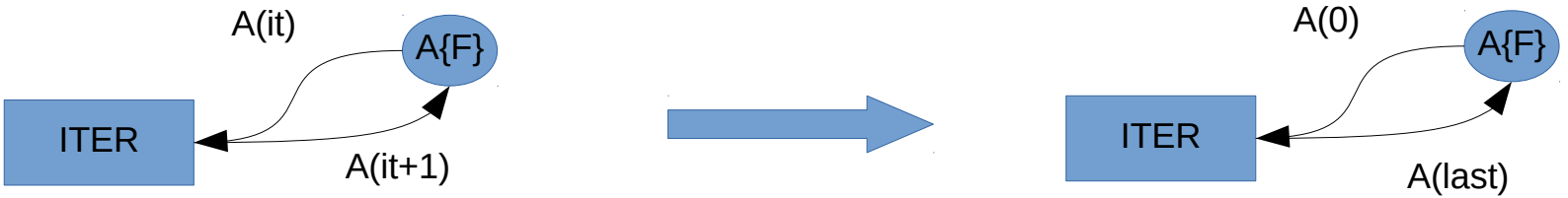
Архитектура компилятора LuNA-2



Обзор основных модулей компилятора LuNA-2

Модули IPP

Luna program:
….
while (it < 500), it = 0 .. out last : <Fi(it) --> Fi(it+1)>
{….}
….



Модуль IBC

- Назначение: построение прямого управления на множестве блоков
- Наивный алгоритм (в случае, если возможно исполнения двух блоков в любом порядке, управление задается произвольным образом)

Модуль ВСВ

- Назначение: построение управления на множестве ФВ, входящих в блок
- Прототип выполняет выполнение заданных ограничений, накладываемых на ФА

Модуль RDB

- Назначение: построение распределения ФД и ФВ на узлы мультикомпьютера
- Прототип выполняет выполнение заданных ограничений, накладываемых на ФА (доступ к элементам массива ФД вида M[итератор ФВ типа for], равные размерности массивов, равные размеры массивов ФД по всем измерениям)
- Распределение осуществляется наивным способом: все массивы ФД, входящие в области P2CR, распределяются между узлами мультикомпьютера равными частями, ФВ распределяются исходя из заданного распределения массивов ФД

Модуль FPASM-Gen

- Назначение: преобразование внутреннего представления P2CR в представление FPASM, которое далее используется непосредственно при генерации MPI-программы

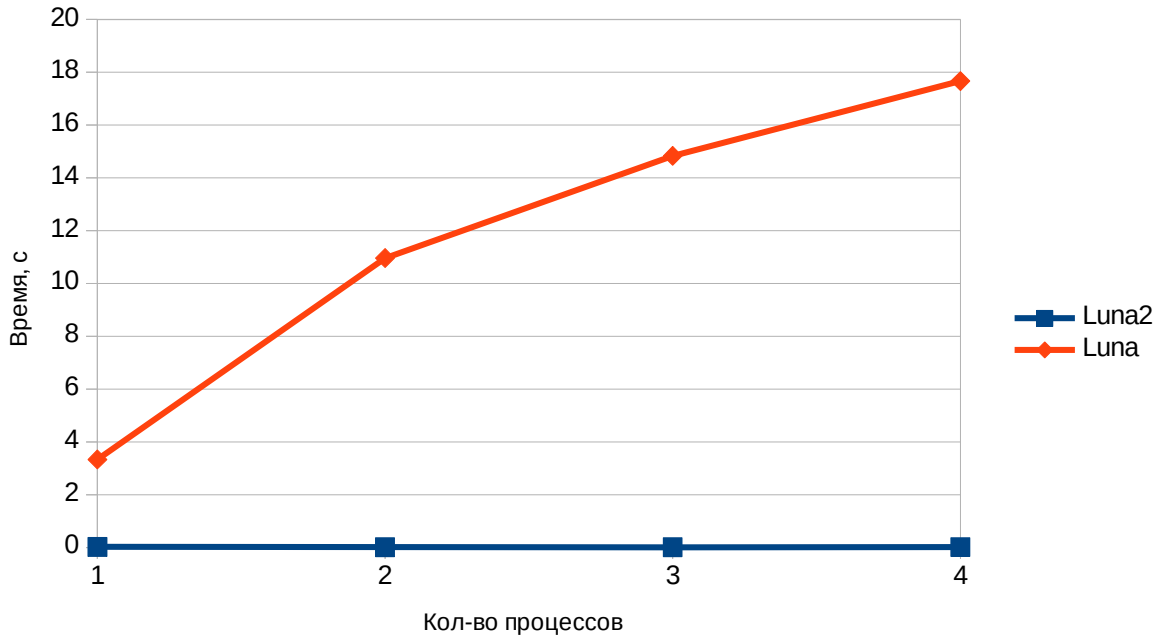
Результаты сравнительных тестов производительности

Исходные данные

- Явная схема (2-D), 1-D фрагментация расчетной области, изменялось кол-во и размер ФД, кол-во итераций
- Сравнивались результаты замеров времени выполнения сгенерированной компилятором LuNA-2 программы и выполнения фрагментированной программы в системе LuNA
- Оборудование: 1 узел, CPU: Intel(R) Core(TM) i7-3820 CPU @ 3.60GHz, 16GB RAM

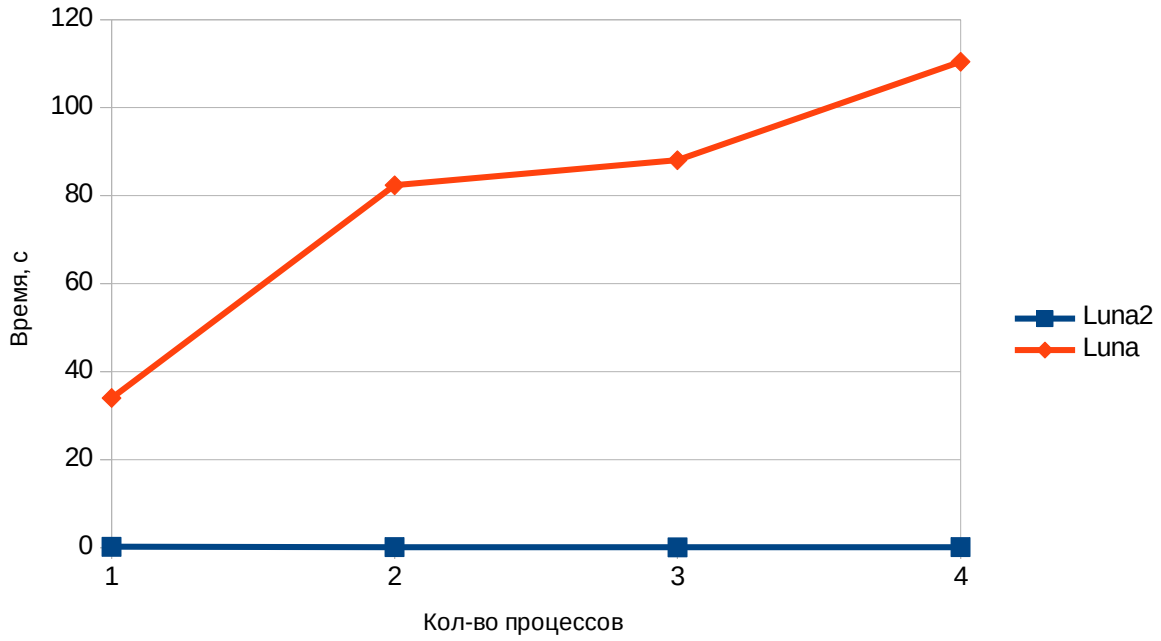
Тест 1

- Область 10 ФД, размер ФД: 1000x1000, 50 итераций



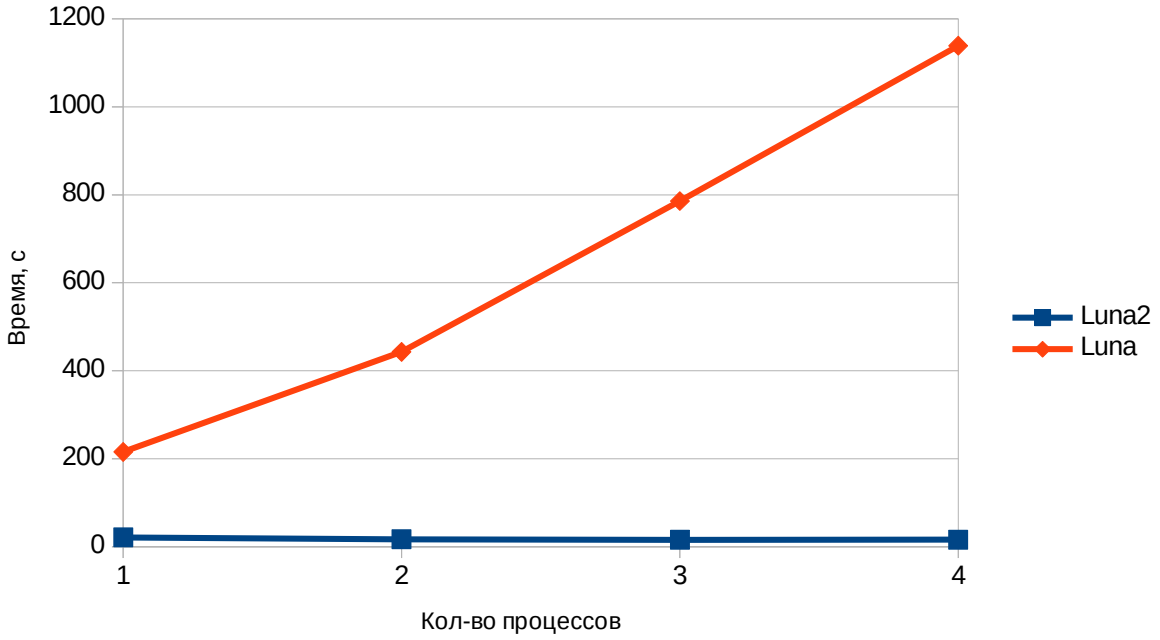
Тест 2

- Область 10 ФД, размер ФД: 1000x1000, 500 итераций



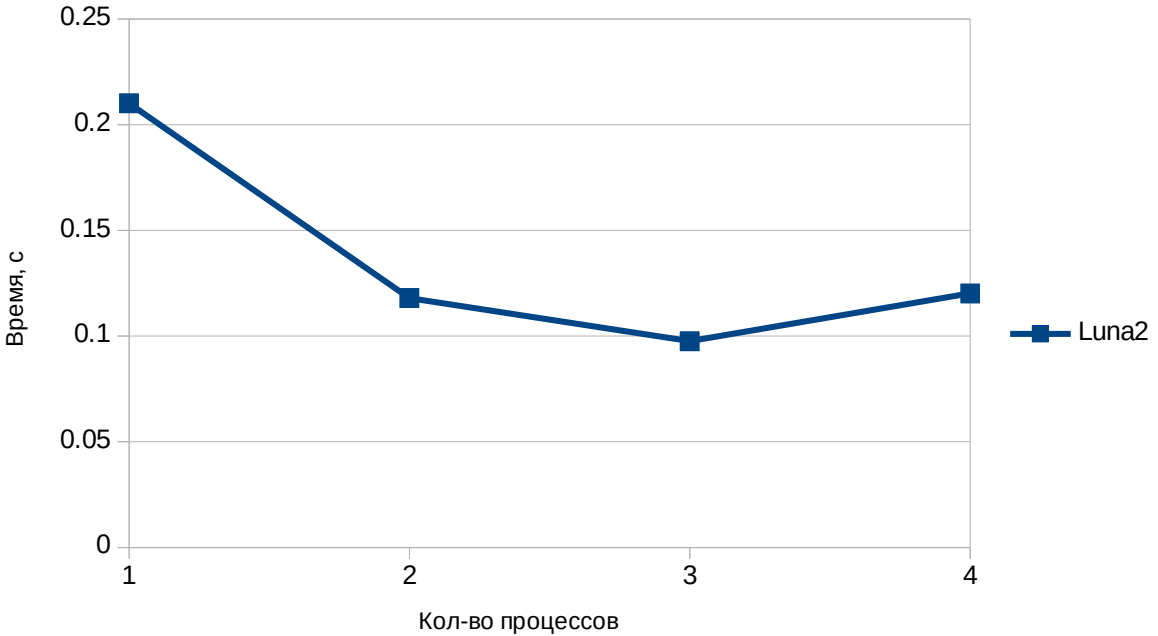
Тест 3

- Область 500 ФД, размер ФД: 10x10, 500 итераций



Тест 4

- Измеряется ускорение, полученное при исполнении сгенерированной программы на нескольких вычислительных ядрах
- Область 10 ФД, размер ФД: 1000x1000, 500 итераций



Текущие результаты

- Разработана архитектура системы LuNA-2
- Разработаны алгоритмы распределения ресурсов и построения прямого управления для ФА, удовлетворяющих ограничениям
- Разработан прототип компилятора LuNA-2
- Проведены сравнительные тесты производительности систем LuNA и LuNA-2 на тестовой задаче

Спасибо за внимание!