Утверждаю  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_ руководитель проекта   
д.ф.-м.н. В.А.Вшивков

# Отчет по проекту РНФ № 16-11-10028 А.В.Снытникова за апрель 2017 года

Аннотация

Статья «**Porting a Plasma Simulation PIC code from CUDA to MIC»** отправлена в журнал Parallel Computing. Проведена оптимизация параллельного алгоритма, позволяющая использовать в расчетах до 600 ядер. Показана эффективность использования асинхронных пересылок. Текст статьи прилагается к отчету. (Где текст статьи?)

Contents

[Отчет по проекту РНФ № 16-11-10028 А.В.Снытникова за апрель 2017 года 1](#_Toc481500159)

[Персональный план 2](#_Toc481500160)

[Отправка статьи 2](#_Toc481500161)

[Описание используемой модели 2](#_Toc481500162)

[Проведение расчётов по моделированию взаимодействия электронных пучков с плазмой 4](#_Toc481500163)

[Расчет на грубой сетке, 80×20×20 узлов. 5](#_Toc481500164)

[Расчет на промежуточной (отладочной) сетке, 400×100×20 узлов. 6](#_Toc481500165)

[Оптимизация параллельного алгоритма расчёта взаимодействия электронных пучков с плазмой 8](#_Toc481500166)

[Оптимизация межпроцессорных пересылок и переход к асинхронным пересылкам в случае необходимости 12](#_Toc481500167)

Персональный план работы А.В.Снытникова на апрель 2017 с соотвествии с планом работ по проекту РНФ № 16-11-10028, согласованный с Руководителем проекта

Оптимизация межпроцессорных пересылок и переход к асинхронным пересылкам в случае необходимости. Оптимизация параллельного алгоритма расчёта взаимодействия электронных пучков с плазмой для выполнения большой серии расчётов на суперкомпьютерах.

Этот план содержит две основных части:

1. Проведение расчётов по моделированию взаимодействия электронных пучков с плазмой
2. Оптимизация параллельного алгоритма расчёта взаимодействия электронных пучков с плазмой
3. Оптимизация межпроцессорных пересылок и переход к асинхронным пересылкам в случае необходимости.

Кроме того, к данному планы были рукописным шрифтом добавлены составление **описания используемой модели** и **отправка одной статьи**.

## Отправка статьи

Статья «**Porting a Plasma Simulation PIC code from CUDA to MIC»**посвященная созданию плазменных кодов, свободно переносимых между различными типами вычислительных ускорителей, отправлена в журнал Parallel Computing (база данных Scopus) .

## Описание используемой модели

Используемая в данной работе модель в основном (кроме граничных условий) следует описанному в статье В.А.Вшивкова и А.В.Снытникова «**ВЫЧИСЛЕНИЕ ТЕМПЕРАТУРЫ ПРИ МОДЕЛИРОВАНИИ ВЫСОКОТЕМПЕРАТУРНОЙ ПЛАЗМЫ МЕТОДОМ ЧАСТИЦ-В-ЯЧЕЙКАХ НА СУПЕРЭВМ**», Научный Вестник НГТУ, 2010.

Численная модель высокотемпературной бесстолкновительной плазмы состоит из уравнения Власова и системы уравнений Максвелла:

.

Не приведены формулы для **j** и ρ.

Плотность заряда и тока вычисляются по формулам:

Где n – полное число частиц, xk, qk, vk – координата, заряд и скорость частицы с номером *k*. Функция R(x) – форм-фактор (ядро) иодельной частицы.

В данной работе используется процедура решения этих уравнений, описанная в работе (В.А.Вшивков, А.В.Снытников, Научный Вестник НГТУ, 2010). (?) Далее все уравнения будут приводится в безразмерном виде. Для обезразмеривания используются следующие базовые величины:

* скорость света c = 3×1010 см/с
* масса электрона me = 9.1×10-31 кг
* заряд электрона e = 1.6×10-19 Кл
* плотность плазмы n0 = 1014 см-3
* плазменная электронная частота e = 5.6×104 n1/2 c-1
* для нормировки значений электрического и магнитного полей используется величина E0 = mee c/2e

(На что нормируются электрическое и магнитное поля?)

Уравнение Власова решается методом частиц в ячейках. В этом методе вся плазма моделируется набором отдельных частиц, каждая из которых характеризует движение многих физических частиц. Характеристики уравнения Власова описывают траектории этих частиц. Уравнения этих характеристик имеют вид для электронов:

.

И для ионов:

Здесь коэффициент ** для электронов равен 1, а для ионов - отношению масс электрона и иона (?).

Для нахождения электрических и магнитных полей используется схема Лэнгдона-Лазински, в которой поля определяются из разностных аналогов законов Фарадея и Ампера [4]: (?)

.

(Какая схема? Обозначения?)здесь roth – сеточный эквивалент ротора, n – номер временного шага.

Эта схема имеет второй порядок аппроксимации по пространству и по времени.

Задача ставится следующим образом. В начальный момент в трехмерной области решения, которая имеет форму прямоугольного параллелепипеда: (или куба?)

*0 ≤ x ≤ Lx, 0 ≤ y ≤ Ly, 0 ≤ z ≤ Lz*,

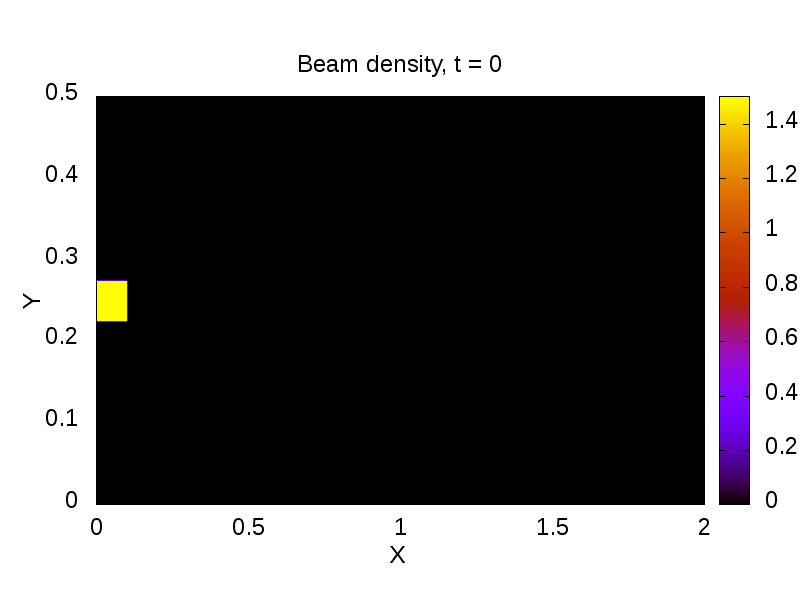
находится плазма, состоящая из электронов и ионов. Частицы (как реальные, так и соответствующие им модельные) распределены по области квазиравномерно, т.е. координаты задаются с помощью генератора случайных чисел с равномерным распределением. Плазма занимает часть объема области, размер части области, зпнятой плазмой, задается отдельно, в конфигурационном файле. В частности, задается ширина области, занятой плазмой: Dy и Dz. В таком случае частицы плазмы распределяются в интервале:

0 ≤ x ≤ Lx,

Ly/2 - Dy/2 ≤ y ≤ Ly/2 + Dy/2,

Lz/2 - Dz/2 ≤ z ≤ Lz/2 + Dz/2

(Как реализована равномерность распределения?). Задается плотность плазмы и температура электронов (как задаются необходимые плотность и температура?), температура ионов считается нулевой. Температура электронов задается как параметр генератора случайных чисел с нормальным распределением. Масса иона равна 1836 масс электрона (ион водорода), заряд иона равен +1.



**Рис. 1.** Плотность пучка в начальный момент времени. Размер входного буфера для пучка, подробно описан ниже:Px = 0.2. Размер области Lx = 2, Ly = 0.5, Lx = 0.05. Размер сетки 800×200×20, таким образом шаги сетки одинаковы по всем направлениям(Видимо, шаги сетки по направлениям сильно отличаются, почему?), 100 частиц каждого типа в ячейке (Как распределяются частицы?) описано выше, размер области показан на рисунке, z=0.025. (при каком *z*?)

### Описание входного буфера для пучка

Пучок вводится в область с помощью входного буфера (?), находящего внутри области, как показано на рисунке 1, желтый цвет показывает подобласть, изначально заполненную пучком. В конфигурационном файле задается изначальный размер пучка: Px, Py, Pz, далее частицы пучка распределяются и интервале:

0 ≤ xb ≤ Px,

Ly/2 - Py/2 ≤ yb ≤ Ly/2 + Py/2,

Lz/2 - Pz/2 ≤ zb ≤ Lz/2 + Pz/2

Импульс частиц пучка имеет вид: pb = (p,0,0), т.е. частицы пучка движутся строго по X. Тепловая добавка к импульсу по X и Y является малой и в данном разделе для простоты не рассматривается. Внутри буфера задаются условия компенсации по току, с тем, чтобы ток входящих частиц (пучка) компенсировался противотоком электронов, и не создавалось нефизических эффектов на границе. Таким образом, если скорость частицы-электрона пучка внутри равна v, то создается две частицы-электрона плазмы (каждая со вдвое меньшим зарядом, чем для пучковой частицы)

Для частицы-электрона пучка: координаты (xb,yb,zb), импульс pb, скорость vb = p/

Для частиц-электронов плазмы: координаты (xb,yb,zb), скорость vp1,2 = -p/vT1,2

Здесь vT1,2 –тепловая добавка к скорости электронов плазмы, которая как раз и обеспечивает заранее задаваемую температуру. Таким образом видно, что суммарный ток описанных выше частицы пучка и двух частиц-электронов плазмы сводится к тепловому току электронов. Таково изначальное распределение частиц в буферной области.

По мере пересечения частицами пучка правой границы буфера (Px), в буферную область добавляются новые частицы-электроны пучка c с координатами (xb,yb,zb) задаеваемыми случайными числами с равномерным распределением в интервале:

0 ≤ xb ≤ Px,

Ly/2 - Py/2 ≤ yb ≤ Ly/2 + Py/2,

Lz/2 - Pz/2 ≤ zb ≤ Lz/2 + Pz/2

и вместе с ними добавляются новые добавочные частицы-электроны плазмы с параметрами, описанными выше, для обеспечения токовой компенсации на границе. Таким образом организуется непрерывное введение пучка в область.

Граничные условия периодические по координатам Z и Y1 и **непериодические по X:** условия равенства нулю нормальной производной, например, для электрического поля.

Или, в разностном варианте, при числе узлов в области Nx и при нумерации узлов с 0:

E0 = E1, E Nx = E Nx-1

Аналогично для остальных компонент электрического и магнитного поля и для токов.(Какие граничные условия на границах по х и как реализуются?)

## Проведение расчётов по моделированию взаимодействия электронных пучков с плазмой

В этом разделе показаны начальные стадии расчета по введению пучка в область с целью показать наличие либо отсутствие краевых эффектов в области ввода пучка, а также экспериментального подбора величины сетки, необходимого числа процессоров и времени счета. Кроме того, в соответствии с одной из двух основных задач этого месяца, «Оптимизация параллельного алгоритма...» необходимо было выяснить наличие потребности в оптимизации, т.е. насколько эффективно или неэффективно реализован параллельный алгоритм на данный момент.

### Расчет на грубой сетке, 80×20×20 узлов.

Расчеты по моделированию взаимодействия электронных пучков с плазмой (Какие расчёты?) проводились на двух сетках: 80×20×20 узлов и 400×100×20 узлов. Несмотря на то, что первая из них является очевидно слишком грубой, целесообразно провести расчет в том числе и на такой сетке для выяснения наличия краевых эффектов. На рисунках 2-4 видно, что на левой границе области не возникает резких изменений электрического поля (?), которые были бы неизбежны при наличии тока через границу.

|  |  |
| --- | --- |
|  |  |
| Рис. 2а. Плотность электронов пучка, нормированная на среднюю плотность электронов пучка в начальный момент времени | Рис. 2б. Электрическое поле ***Ex*** |
| Рис. 2. момент времени t =0.1. Сетка 80×20×20, 100 частиц каждого типа в ячейке, Размер области 1×0.25×0.25, 2 процессорных ядра. Расчет проведен на кластере НГУ. | |

|  |  |
| --- | --- |
|  |  |
| Рис. 3а. Плотность электронов пучка, нормированная на среднюю плотность электронов пучка в начальный момент времени | Рис. 3б. Электрическое поле ***Ex*** |
| Рис. 3. момент времени t =0.2. Сетка 80×20×20, 100 частиц каждого типа в ячейке, Размер области 1×0.25×0.25, 2 процессорных ядра. Расчет проведен на кластере НГУ. | |

|  |  |
| --- | --- |
|  |  |
| Рис. 4а. Плотность электронов пучка, нормированная на среднюю плотность электронов пучка в начальный момент времени | Рис. 4б. Электрическое поле ***Ex*** |
| Рис. 4. момент времени t =0.1. Сетка 80×20×20, 100 частиц каждого типа в ячейке, Размер области 1×0.25×0.25, 2 процессорных ядра. Расчет проведен на кластере НГУ. | |

(Не понятно, что показывают эти рисунки.) Из рисунков видно, что использованная сетка является очевидно недостаточной для моделирования, но, поскольку это фактический максимум того, что может быть просчитано на однопроцессорной машине, из этого следует необходимость распаралеливания и счета на суперЭВМ.

### Расчет на промежуточной (отладочной) сетке, 400×100×20 узлов.

Далее, проведены расчеты на сетке 400×100×20, имеющей промежуточный характер, для тех же расчетных параметров, что и в предыдущем разделе (Задача та же?) На рисунках 5б-7б видно, что на такой сетке электрическое поле имеет качественно правильную структуру, т.е. ненулевые значения поля присутствуют только в области, занятой пучком и по мере продвижения пучка вознивает чередующаяся последовательность минимумов и максимумов. (?)

Проводились также расчета на сетке 800×800×20, т.е. с размерностью близкой к тому, что используется в работах И.В.Тимофеева, но в данном отчете они не приводятся. (?)

|  |  |
| --- | --- |
|  |  |
| Рис. 5а. Плотность электронов пучка, нормированная на среднюю плотность электронов пучка в начальный момент времени | Рис. 5б. Электрическое поле ***Ex*** |
| Рис. 5. момент времени t =0.01. Сетка 400×100×20, 100 частиц каждого типа в ячейке, Размер области 8×2×0.2, 8 процессорных ядер. Расчет проведен на кластере НГУ. | |

|  |  |
| --- | --- |
|  |  |
| Рис. 6а. Плотность электронов пучка, нормированная на среднюю плотность электронов пучка в начальный момент времени | Рис. 6б. Электрическое поле ***Ex*** |
| Рис. 6. момент времени t =0.02. Сетка 400×100×20, 100 частиц каждого типа в ячейке, Размер области 8×2×0.2, 8 процессорных ядер. Расчет проведен на кластере НГУ. | |

|  |  |
| --- | --- |
|  |  |
| Рис. 7а. Плотность электронов пучка, нормированная на среднюю плотность электронов пучка в начальный момент времени | Рис. 7б. Электрическое поле ***Ex*** |
| Рис. 7. момент времени t =0.01. Сетка 400×100×20, 100 частиц каждого типа в ячейке, Размер области 8×2×0.2, 8 процессорных ядер. Расчет проведен на кластере НГУ. | |

(Рисунки сильно отличаются. Какой вывод?) На данный момент вывод сделать нельзя, необходимо продолжить расчет.

Для выяснения эффективности распараллеливания была проведена профилировка программы с помощью инструмента Intel Trace Analyzer&Collector (рис. 8). На рисунке видно, что большую часть времени программа проводит в коммуникациях, конкретно в процессе выполнения операции MPI\_Allreduce (сложение трехмерных матриц токов). Таким образом необходимость «Оптимизация параллельного алгоритма...» очевидна. (Ни разу не показана необходимость распараллеливания и использования суперкомпьютеров) описано в конце предыдущего подраздела.

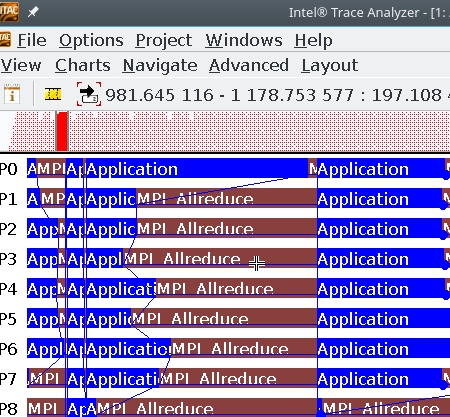


Рис.8. Результат профилировки программы с помощью Intel Trace Analyzer&Collector на кластере НГУ. Горизонтальные линии(трассы) означают отдельные параллельные процессы, тонкие линии между трассами – межпроцессорные коммуникации, номера процессов показаны слева (от 0 до 8, всего 32 процесса). Красный цвет трассы означает пересылки, синий-счет. Показан отдельный интервал расчета на сетке 800×800×20, на 32 процессорах. (На рисунке ничего не видно) рисунок отмасштабирован

## Оптимизация параллельного алгоритма расчёта взаимодействия электронных пучков с плазмой

С учетом представленного на рисунке 8 (преобладание коллективных коммуникаций, операция MPI\_Allreduce) была проведена оптимизация параллельного алгоритма, которая заключается в следующем:

* Замена реализации MPI на более новую
* Уменьшение количества пересылаемых данных
* Исключение промежуточных внешних (файловых) операций ввода-вывода
* Уменьшение количества процессов, участвующих в каждой коллективной операции
* Увеличение количества частиц в ячейке (**полное количество частиц меняется с увеличением числа процессоров**)

(Не означает ли это, что первоначальная программа была написана плохо, поэтому потребовалась оптимизация?) Из того факта, что необходимо было проведение оптимизации, не следует, что исходная программа была написана неправильно, ибо исходная программа создавалась на другой технике и для других задач, с другой спецификой межпроцессорных обменов.

Результат оптимизации программы по моделированию взаимодействия электронных пучков с плазмой показан на рисунках (Для какой задачи? Или вообще другой?)

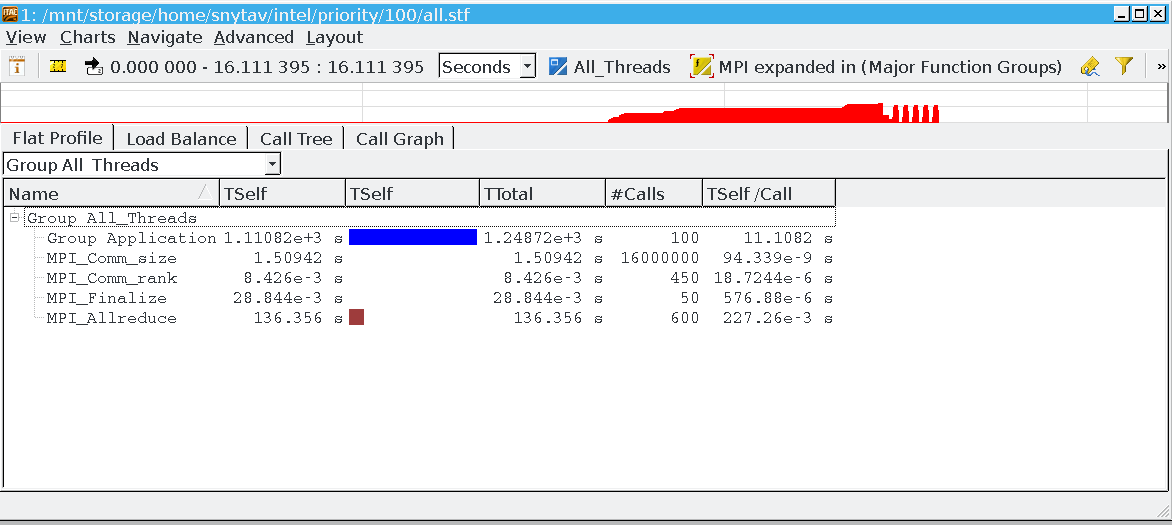


Рис. 9. Время выполнения коммуникационных операций (названия операций даны на рисунке, время показано красным цветов в графе TSelf) в программе в сравнении с временем вычислений (Group Application, синий цвет в графе TSelf). Расчет на сетке 1000×1000×4 узла,10 временных шагов, 100 частиц в ячейке, **100 процессорных ядер**. Результат профилировки программы с помощью Intel Trace Analyzer&Collector на кластере НГУ.

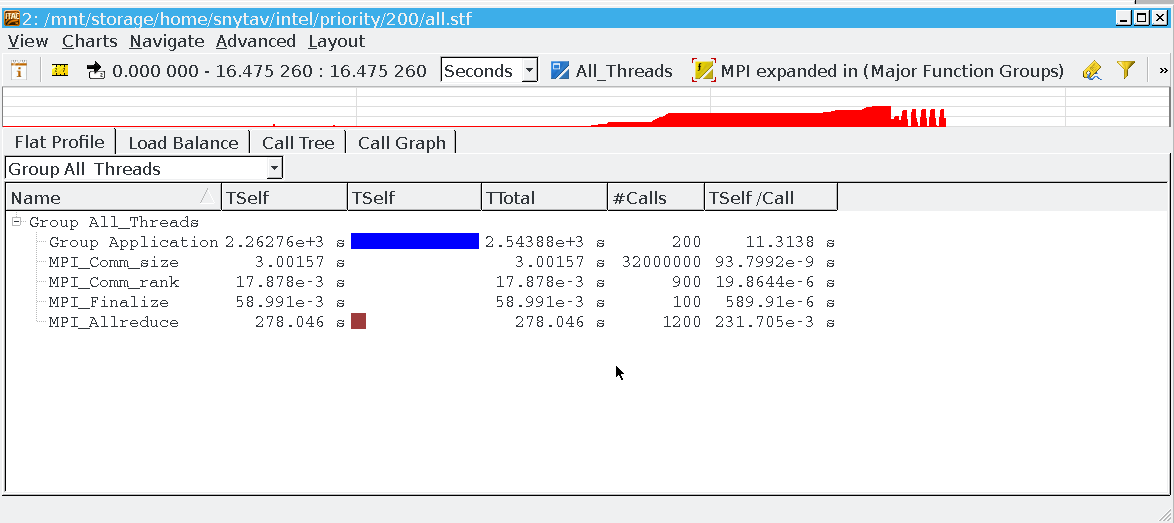


Рис. 11. Время выполнения коммуникационных операций (названия операций даны на рисунке, время показано красным цветов в графе TSelf) в программе в сравнении с временем вычислений (Group Application, синий цвет в графе TSelf). Расчет на сетке 1000×1000×4 узла, 200 частиц в ячейке, **200 процессорных ядер**. Результат профилировки программы с помощью Intel Trace Analyzer&Collector на кластере НГУ.

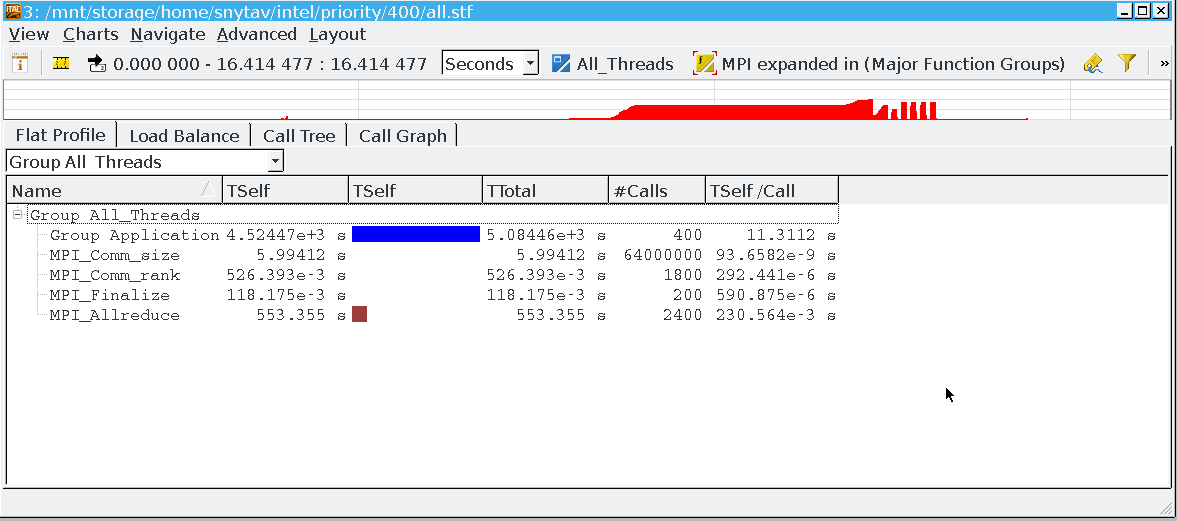


Рис. 11. Время выполнения коммуникационных операций (названия операций даны на рисунке, время показано красным цветов в графе TSelf) в программе в сравнении с временем вычислений (Group Application, синий цвет в графе TSelf). Расчет на сетке 1000×1000×4 узла, 400 частиц в ячейке, **400 процессорных ядер**. Результат профилировки программы с помощью Intel Trace Analyzer&Collector на кластере НГУ.

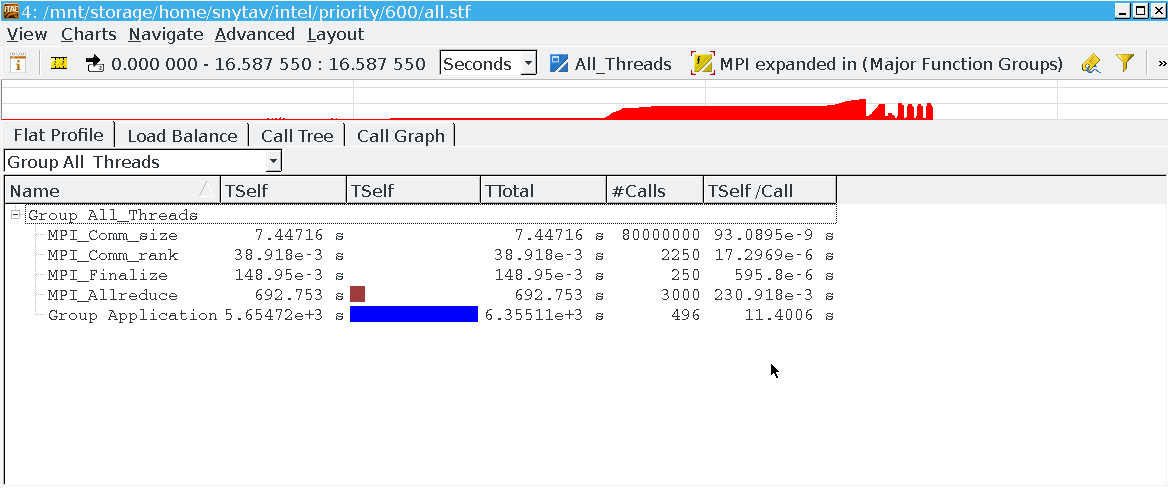


Рис. 12. Время выполнения коммуникационных операций (названия операций даны на рисунке, время показано красным цветов в графе TSelf) в программе в сравнении с временем вычислений (Group Application, синий цвет в графе TSelf). Расчет на сетке 1000×1000×4 узла, 600 частиц в ячейке, **600 процессорных ядер**. Результат профилировки программы с помощью Intel Trace Analyzer&Collector на кластере НГУ.

Сводя полученные результаты измерения времени воедино, получаем таблицу 1.

**Таблица 1.** Время вычислений и время комуникаций в зависимости от числа процессов.

|  |  |  |  |
| --- | --- | --- | --- |
| Число процессорных ядер | Число частиц в ячейке | Время вычислений (суммарное по всем процессам) | Время коммуникаций |
| 100 | 100 | 1248 | 136 |
| 200 | 200 | 2543 | 278 |
| 400 | 400 | 5084 | 553 |
| 600 | 600 | 6335 сек. | 692.7 сек. |

Относительная часть времени вычислений от общего времени работы программы (при растущем вместе с числом процессоров объеме вычислений) дает величину эффективности распараллеливания в слабом смысле, которая показана на рисунке 13.

Рис. 13. Эффективность распараллеливания в слабом смысле. Расчет на сетке 1000×1000×4 узла. Измерение времени проведено с помощью Intel Trace Analyzer&Collector на кластере НГУ.

Достаточно высокая эффективность даже на 600 ядрах показывет, что

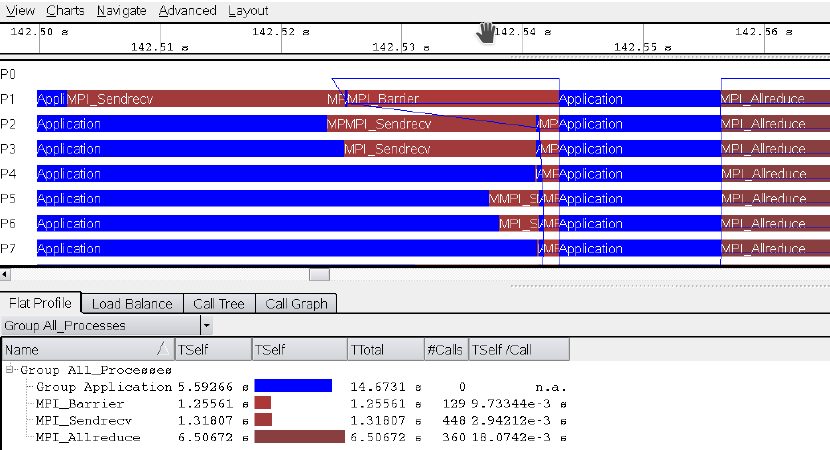
1. Проведенная оптимизация была успешной
2. Программа готова к расчетам на большом количестве ядер

(Достаточно ли этой оптимизации для решения задач по проекту? Какие максимальные возможности? Для какого кластера? На каком кластере планируются вычисления?)

Расчеты планируется проводить на кластере НГУ, на кластере НКС-1П (ИВМиМГ СО РАН), на «Ломоносов» (НИВЦ МГУ) и на кластере «Политехник» (СПбГТУ). Прояснение возможностей программы для других кластеров запланировано на май 2017. Исходя из рисунков 11-12 можно предположить, что потребуется дополнительная оптимизация для работы на 2000 ядер и более, впрочем на данный момент нет уверенности, что это необходимо для проекта.

## Оптимизация межпроцессорных пересылок и переход к асинхронным пересылкам в случае необходимости

На рисунке 14 показан этап работы параллельной программы по моделированию взаимодействия электронных пучков с плазмой (Какой программы? Для решения какой задачи?), соответствующий решению уравнений Максвелла. В данном случае расчетная область разделена по координате Y и для обмена граничными значениями используется процедуры парного межпроцессорного обмена библиотеки MPI (MPI\_Sendrecv).



**Рис.14.** Результат профилировки программы с помощью Intel Trace Analyzer&Collector на кластере НКС-30Т, ИВМиМГ СО РАН. Горизонтальные линии(трассы) означают отдельные параллельные процессы, тонкие линии между трассами – межпроцессорные коммуникации, номера процессов показаны слева (от 0 до 7, всего 8 процессов). Красный цвет трассы означает пересылки, синий-счет. Показан этап расчета, соответствующий решению уравнений Максвелла, сетка 800х800х20.

Из рисунка 14 видно, что в течение временного интервала решения уравнений Максвелла некоторые процессы (на рисунке P1, P2,P3,P5) проводят много времени в состоянии ожидания отправки сообщений, т.е. ждут пока процесс-приемник будет готов принять сообщение. Использование асинхронных коммуникаций (MPI\_Isendrecv) может значительно сократить это время. Следует отметить, что до нуля все же время сократить невозможно, так как в используемом численном методе (схема Лэнгдона-Лазински) присутствует зависимость по данным, и процесс номер, например, 6, вынужден будет все же дождаться, пока процесс номер 5 пришлет ему свои граничные значения (или нужно менять численный метод, что в данной работе не рассматривается).

Эффективность распараллеливания в данном случае можно измерить следующим образом. В интервале абсолютного времени от 142.50 сек. до 142.555 секунды, пока решаются уравнения Максвелла, рассматриваемые процессы тратят на вычисления соответсвенно:

**Таблица 2.** Время, затраченнное на вычисления в ходе параллельного решения уравнений Максвелла, , сетка 800х800х20, 8 процессов, при использовании синхронных пересылок.

|  |  |  |
| --- | --- | --- |
| Номер  процесса | Время на вычисления, сек. | Время на коммуникации, (синхронные пересылки), сек. |
| 0 | 0.003 | 0.052 |
| 1 | 0.025 | 0.03 |
| 2 | 0.027 | 0.028 |
| 3 | 0.053 | 0.002 |
| 4 | 0.052 | 0.003 |
| 5 | 0.049 | 0.006 |
| 6 | 0.050 | 0.005 |
| 7 | 0.052 | 0.003 |

Синхронные пересылки MPI\_Sendrecv были заменены на асинхронные (MPI\_Isendrecv), в результате общее время решения уравнений Максвелла снизилось до 0.045 сек., а время, затраченное на пересылки, изменилось, как показано в таблице 3

**Таблица 2.** Время, затраченнное на вычисления в ходе параллельного решения уравнений Максвелла, , сетка 800х800х20, 8 процессов, при использовании асинхронных пересылок.

|  |  |  |
| --- | --- | --- |
| Номер  процесса | Время на вычисления, сек. | Время на коммуникации, (синхронные пересылки), сек. |
| 0 | 0.043 | 0.002 |
| 1 | 0.015 | 0.03 |
| 2 | 0.026 | 0.0145 |
| 3 | 0.0389 | 0.01 |
| 4 | 0.042 | 0.003 |
| 5 | 0.035 | 0.006 |
| 6 | 0.04 | 0.005 |
| 7 | 0.042 | 0.003 |

Для исследования параллельной эффективности данных в таблицах 2 и 3 недостаточно, но во всяком случае то, что общее время решения уравнений Максвелла снизилось на 20 %, позволяет сделать **вывод об эффективности реализованных асинхронных пересылок при использовании декомпозиции области**.

(Значит ли это, что программа всё ещё не готова для решения физических задач?)  
Этот вывод можно сделать по разделу «Проведение расчётов по моделированию взаимодействия электронных пучков с плазмой», но не по текущему разделу.