# Отчет по проекту РНФ 16-11-10028 за 2017 год

**Исполнителя Снытникова А.В.**

Выдержка из отчета за 2016 год

**пункт 3.2. Ожидаемые в конце 2017 года конкретные научные результаты** в части, касающейся Снытникова А.В.

*3) Будет* ***оптимизирован параллельный алгоритм расчёта*** *взаимодействия электронных пучков с плазмой для выполнения большой серии расчётов на суперкомпьютерах. Будут реализованы несколько вариантов распределения вычислительной нагрузки между процессорами, основанные на разделение частиц и эйлерово-лагранжевой декомпозиции, получены* ***оценки эффективности распараллеливания и масштабируемости разработанных алгоритмов*** *для вычислительных комплексов современной архитектуры – «Ломоносов» (МГУ), НКС-30Т (ССКЦ СО РАН) и суперкомпьютер НГУ (Новосибирск).*

*Берендеев*

*Снытников*

*4) Будут проведены* ***расчёты на суперкомпьютере «Ломоносов» (МГУ) с использованием нескольких тысяч вычислительных ядер для числа частиц порядка 10^10-10^11 и размером сетки 10^8-10^10 узлов*** *и получена зависимость точности решения от счётных параметров при большом числе модельных частиц.*

Для высокопроизводительного моделирования турбулентных режимов пучково- плазменного взаимодействия *будет проведена* ***оптимизация мелкозернистого распараллеливания*** *для графических ускорителей Intel Xeon Phi, сделан подбор оптимального набора директив OpenMP и опций компилятора, а также выполнена оптимизация межпроцессорных пересылок и* ***переход к асинхронным пересылкам*** *в случае необходимости.*

*Снытников*

Далее будет описана реализация этих пунктов плана. Выделенные жирным шрифтом формулировки из плана представляют собой заголовки разделов отчета в оглавнении, приведенном ниже:

Оглавление

[Отчет по проекту РНФ 16-11-10028 за 2017 год 1](#_Toc498744574)

[Выдержка из отчета за 2016 год 1](#_Toc498744575)

[Оптимизация параллельного алгоритма. Оценки эффективности распараллеливания и масштабируемости разработанных алгоритмов. Расчеты на суперкомьютере «Ломоносов» для большого числа частиц, порядка 10^10-10^11 2](#_Toc498744576)

[План 2](#_Toc498744577)

[Фактически сделано 3](#_Toc498744578)

[Подробности реализации 3](#_Toc498744579)

[Краткое описание программы 8](#_Toc498744580)

[Входные данные: 8](#_Toc498744581)

[Выходные данные 8](#_Toc498744582)

[Техническое описание 8](#_Toc498744583)

[Оптимизация мелкозернистого распараллеливания 9](#_Toc498744584)

[План 9](#_Toc498744585)

[Фактически сделано 9](#_Toc498744586)

[Подбор оптимального набора директив OpenMP и опций компилятора 11](#_Toc498744587)

[Вывод по третьей части отчета: 11](#_Toc498744588)

[Переход к асинхронным пересылкам 11](#_Toc498744589)

[План 11](#_Toc498744590)

[Фактически сделано 11](#_Toc498744591)

[Подробности реализации 11](#_Toc498744592)

[Литература 13](#_Toc498744593)

## Оптимизация параллельного алгоритма. Оценки эффективности распараллеливания и масштабируемости разработанных алгоритмов. Расчеты на суперкомпьютере «Ломоносов» для большого числа частиц, порядка 10^10-10^11

### План

3) *Будет оптимизирован параллельный алгоритм расчёта взаимодействия электронных пучков с плазмой для выполнения большой серии расчётов на суперкомпьютерах. Будут реализованы несколько вариантов распределения вычислительной нагрузки между процессорами, основанные на разделение частиц и эйлерово-лагранжевой декомпозиции, получены оценки эффективности распараллеливания и масштабируемости разработанных алгоритмов для вычислительных комплексов современной архитектуры – «Ломоносов» (МГУ), НКС-30Т (ССКЦ СО РАН) и суперкомпьютер НГУ (Новосибирск).*

*Будут проведены расчёты на суперкомпьютере «Ломоносов» (МГУ) с использованием нескольких тысяч вычислительных ядер для числа частиц порядка 10^10-10^11 и размером сетки 10^8-10^10 узлов и получена зависимость точности решения от счётных параметров при большом числе модельных частиц.*

### Фактически сделано

Проведены тестовые расчеты с целью эффективности распараллеливания и масштабируемости разработанных алгоритмов для моделирования взаимодействия электронного пучка с плазмой, а также с целью выяснения реальных возможностей суперЭВМ по проведению физических расчетов. Показано, что текущая версия программы позволяет проводить расчет на трехмерной сетке размером **3503 узлов при 100 частицах в ячейке за 1 сутки** на 10 ядрах суперкомпьютера «Ломоносов», или 42 млн.узлов. Эффективность распараллеливания составила **92 %** для максимального использованного числа ядер: 100 для кластера НГУ и 200 для кластера «Политехник». Таким образом раработанная программа **может считаться готовой** к проведению запланированных расчетов на сетках размерностью свыше 100 млн. узлов. Результаты опубликованы в статье [1].

Расчёты на суперкомпьютере «Ломоносов» (МГУ), на гибридной его части с использованием нескольких тысяч вычислительных ядер проведены для числа частиц порядка 10^10-10^11 и размером сетки 10^8-10^10 узлов. Результаты опубликованы в тезисах [2]. Использовано в одном расчете до 500 ускорителей Nvidia Tesla, с оперативной памятью 6 Гб, по 125 млн. модельных частиц на каждый ускоритель, всего порядка 50 млрд. модельных частиц, сетка 1000х1000х1000, 500 млн. узлов.

### Подробности реализации

**Целью расчетов**, проведенных в июне 2017 в рамках проекта РНФ № 16-11-10028 является ответ на вопрос, **какие расчеты** (т.е. какой размерности) могут бытть проведены на **доступных суперЭВМ**, на каком количестве процессоров (или процессорных ядер) и за какое время.

Для того, чтобы получить ответ на этот вопрос, было запущено большое количество тестовых расчетов, параметры которых приведены в отчете прошлого месяца. Эти тестовые расчеты носят предварительный характер, и проводятся также с целью выяснения реальных возможностей суперЭВМ для проведения физически содержательных расчетов, выполнение которых запланировано на июль 2017. Расчеты проводились на следующих суперЭВМ: кластер СпбГТУ “Политехник”, кластер НКС-1П (ССКЦ СО РАН), кластер НГУ, суперкомпьютер НИВЦ МГУ “Ломоносов”. Первоначально запланированные расчеты на кластере НКС-30Т (ССКЦ СО РАН) не проводились в силу того, что его архитектура идентична таковой для кластера НГУ, а также технических проблем, возникших на НКС-30Т в июне 2017.

Следует отметить, что значительная часть запущенных расчетов не была успешно завершена – как по субъективным причинам (недостатки разработанной программы), так и по объективным (заполненность очереди задач, низкая производительность коммуникационного оборудования, системные ограничения – например, на количество одновременно запущенных задач, или занятых узлов). Основные параметры успешно проведенных расчетов показаны в таблице 1.

В ходе проведения тестовых расчетов было получено большое количество числовых характеристик – время расчета для различных сеток (таблица 1), продолжительность коммуникационных операций (рис. 2, 3), эффективность в сильном и слабом смысле (рис.3, 4), причем все это как для лагранжевой, так и для эйлеровой и декомпозиции и для четырех различных суперЭВМ.

**Рис. 1.** Оценка **размерности расчета** продолжительностью 10 тыс. временных шагов, **который можно провести за *сутки***

Однако с точки зрения проекта РНФ № 16-11-10028 (с точки зрения проведения реальных физических расчетов) имеет смысл только один параметр – оценка **размерности расчета** продолжительностью 10 тыс. временных шагов, **который можно провести за *сутки*** в данной параллельной конфигурации (при заданном числе процессорных ядер и заданном числе подобластей, на которые разделяется расчетная область), т.е. без увеличения времени коммуникаций. Под **размерностью расчета** понимается **количество узлов *N* трехмерной кубической сетки** (полный размер сетки *N\*N\*N*узлов). Одинаковое количество узлов по каждому направлению взято для простоты представления. Как факт в данном случае имеется время счета одного узла, и распределения количества узлов по направлениям ничего существенно не меняет. вдоль одного измерения, рис.1. Число частиц во всех тестовых расчетах 100 в ячейке, соответственно и при проведении оценок будем исходить из этого. Размерность расчета оценивается следующим образом:

1. за основу берется **продолжительность временного шага**, реально измеренная в ходе тестового расчета
2. вычисляется отношение времени 8.64 сек. (один шаг из 10 тыс., проводимых за сутки) к измеренной продолжительности временного шага, обозначим это отношение ***k***
3. **размер сетки, реально имеющийся** в данном тестовом расчете, умножается на ***k,*** и из произведения извлекается кубический корень
4. в том случае, когда сетка вычисленного размера (вместе с частицами) не помещается в память узла, указывается максимально возможный по объему памяти размер (в основном это касается “Ломоносова”)

Вычисленная таким образом размерность задачи приведена в таблице 1 в выделенной колонке. Кроме того, приведены основные параметры и числовые характеристики тестовых расчетов, обозначенные следующим образом:

* *NX, NY, NZ,* - размер сетки по каждому из измерений
* PALL - общее количество процессорных ядер
* PSUB - число подобластей
* T – длительность тестового расчета (50 временных шагов), сек.
* t - длительность временного шага, сек.
* TMPI\_All - длительность операции MPI\_Allreduce (суммирование токов по всей области), сек.
* TMPI\_Send - длительность операции MPI\_Sendrecv (обмен граничными значениями), сек.

**Таблица 1.** Основные числовые характеристики тестовых расчетов.



**Рис. 2.** Эффективность параллельного выполнения на кластере НКС-1П. На рисунке видно, что доля времени, расходуемого на коллективные операции незначительно уменьшается с ростом числа процессоров.

**Рис. 3.** Уменьшение времени счета одного временного шага с увеличением числа ядер на кластере НГУ.

**Рис.4.** Эффективность распараллеливания, вычисляемая как отношение времени вычислений к полному времени расчета, кластер НГУ.



**Рис.5.** Эффективность распараллеливания, вычисляемая как отношение времени вычислений к полному времени расчета, кластер

## Краткое описание программы

### Входные данные:

* Размер сетки
* Количество процессоров
* Количество частиц в ячейке
* Размер области
* Размер части области, занятой пучком и плазмой
* Внешнее магнитное поле
* Энергия частиц пучка
* Поперечная температура частиц пучка
* Температура плазмы (X,Y,Z)

### Выходные данные

(опционально, каждая выдача может быть отключена)

* Списки модельных частиц (координаты, импульсы)
* Плотности электронов, ионов и электронов пучка
* Электрическое и магнитное поля (все три компоненты)
* Ток (три компоненты)
* Одномерная функция распределения электронов (всех, и пучка, и плазмы) по энергии

### Техническое описание

* Язык программирования: Фортран 90
* Средства распараллеливания: MPI, OpenMP
* Целевая архитектура: суперЭВМ классической архитектуры на основе, в первую очередь, процессоров Intel Xeon и ускорителей Intel Xeon Phi

## Оптимизация мелкозернистого распараллеливания

### План

Для высокопроизводительного моделирования турбулентных режимов пучково- плазменного взаимодействия будет проведена оптимизация мелкозернистого распараллеливания для графических ускорителей Intel Xeon Phi, сделан подбор оптимального набора директив OpenMP и опций компилятора

### Фактически сделано

Для высокопроизводительного моделирования турбулентных режимов пучково-плазменного взаимодействия проведена оптимизация мелкозернистого распараллеливания для ускорителей Intel Xeon Phi, выполнен подбор оптимального набора директив OpenMP и опций компилятора. Результаты опубликованы в статье [3].

Подробности реализации

**Цель работы:** задействовать ускорители Intel Xeon Phi, являющиеся в данный момент наиболее эффективным (из доступных) инструментов решения вычислительных задач, в том числе именно на основе метода частиц для достижения целей Проекта.  
**смысл:** Ускоритель Intel Xeon Phi является супер-многоядерным вычислительным устройством (в текущем варианте 72 ядра, или 288 одновременно исполняемых потоков), таким образом видно, что такая архитектура хорошо подходит для расчетов по методу частиц в ячейках.  
**связь с проектом:** в заявке проекта фигурируют современные гибридные суперЭВМ, это именно они (№ 2, в частности, в текущем списке Топ500).

Под мелкозернистым распараллеливанием подразумевается разделение расчета между параллельными процессами по частицам. До 2017 года в разрабатываемой программе существовала возможность распараллеливания с использованием технологии MPI, т.е. для систем с разделенной памятью. Это можно также считать мелкозернистым распараллеливанием, в силу того, что по процессам распределяются частицы в рамках одного массива.

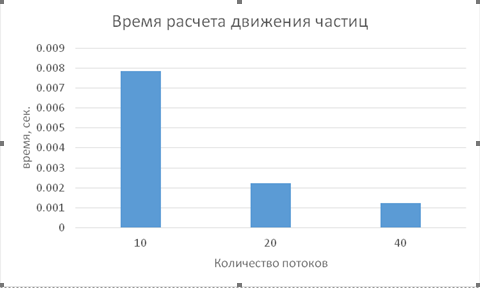
В 2017, в связи с появлением в широком доступе, в том числе в ИВМиМГ гибридных систем, т.е. суперЭВМ с узлами на основе Intel Xeon Phi, т.е. с общей памятью, становится актуальным распараллеливание также и с использованием технологии OpenMP. Такая модель распараллеливания (мелкозернистое распараллеливание) также была реализована. (Подробности – использованы директивы OpenMP в наиболее времяемком фрагменте программы, в цикле движения модельных частиц) Таким образом, речь идет об улучшении (оптимизации) мелкозернистого распараллеливания, которая заключается в подборе оптимального варианта использования средств OpenMP.

В таблице 1 показано время расчета для одного, двух и четырех MPI-процессов, частицы в котором дополнительно разделены между несколькими потоками OpenMP. Здесь важно заметить, что разделение частиц между более чем четырьмя процессами над общей памятью, безусловно, возможно и средствами MPI, без привлечения дополнительных программных средств, и с той же или чуть меньшей эффективностью. Также в таблицу 1 видно, что дальнейшее увеличение числа потоков и процессов не имеет смысла, и дальнейшее повышение скорости счета возможно лишь при использовании новых технологий. Показанные расчеты проведены на процессоре Intel Xeon X5560. Тестирование проводилось на задаче взаимодействия электронного пуска с плазмой после вхождения пучка в область, т.е. при максимальной нагрузке по частицам.

Основной целью данного раздела является создание возможности эффективного использования ускорителей Intel Xeon Phi, что невозможно иначе как с помощью OpenMP. На рисунке 6 показано время расчета на ускорителе Intel Xeon Phi для различного количества потоков. Процессор Intel Xeon X5560 имеет 4 ядра, что обеспечивает эффективный запуск не более чем 6-8 потоков одновременно. Ускоритель Intel Xeon Phi является супер-многоядерным вычислительным устройством (в текущем варианте 72 ядра, или 288 одновременно исполняемых потоков

**Таблица 2.** Время расчета движения частиц (32 тыс.) для нескольких MPI-процессов, частицы в которых дополнительно разделены между несколькими потоками OpenMP на процессоре Intel Xeon.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 MPI-процесс | 2 MPI-процесса | 4 MPI-процесса |
| 1 поток | 0.028 сек. | 0.014 | 0.011 |
| 2 потока | 0.01515 | 0.086 | 0.09 |
| 4 потока | 0.012 | 0.008 | 0.015 |



**Рис.6.** Время расчета движения частиц (32 тыс.) на ускорителе Intel Xeon Phi.

В свете описанного выше различия между Intel Xeon и Intel Xeon Phi (по возможному количеству запускаемых потоков) запустить и там, и там однинаковое количество потоков нельзя (4 потока на Intel Xeon и 4 потока на Intel Xeon Phi – нет смысла, известно, что на Intel Xeon Phi ядра более слабые, 40 потоков на Intel Xeon запустить нельзя, можно сравнивать только в целом, процессор Intel Xeon и ускоритель Intel Xeon Phi). Сравнение в данном случае можно проводить только по итоговому времени.

Выводы по второму разделу: проведена оптимизация мелкозернистого распараллеливания, показана возможность повышения скорости расчета разработанного кода с использованием ускорителей Intel Xeon Phi. 0.01 сек. в таблице (Intel Xeon) против 0.001 сек. на графике (Intel Xeon Phi) т.е. на порядок.

## Подбор оптимального набора директив OpenMP и опций компилятора

Проведенная в данном разделе работа преследует две цели: повысить скорость работы без изменения кода и снять ограничения на размер массивов (частиц и сетки), которые накладывает использование OpenMP. В таблице 2 показаны использованные опции и результат подбора, а именно время работы наиболее затратной части (движения модельных частиц). Оптимизация проводилась на задаче взаимодействия электронного пуска с плазмой после вхождения пучка в область. **Цель проведения оптимизации:** с максимальной эффективностью задействовать ускорители Intel Xeon Phi, являющиеся в данный момент наиболее эффективным (из доступных) инструментов решения вычислительных задач, в том числе именно на основе метода частиц для достижения целей Проекта

**Таблица 3.** Время работы (в мс) процедуры расчета движения модельных частиц.

|  |  |
| --- | --- |
| Основной вариант (-О1 включен по умолчанию) | **155.3 мс** |
| **-O2** | **80.01** |
| **-O3** | **128.029** |
| -O2 -xCORE-AVX2 | **76** |

Вывод по третьей части отчета: наиболее оптимальным является сочетание опций -O2 -xCORE-AVX2 (для Intel Xeon Phi эта опция имеет вид -xCORE-AVX512).

## Переход к асинхронным пересылкам

### План

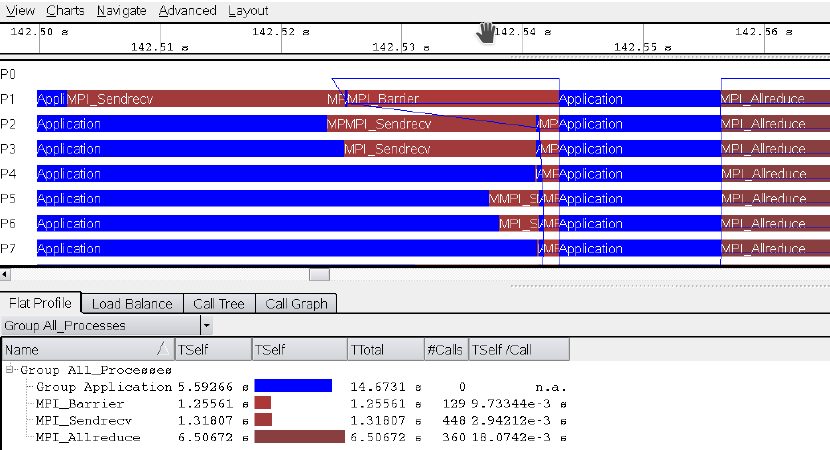
*а также выполнена оптимизация межпроцессорных пересылок и переход к асинхронным пересылкам в случае необходимости.*

### Фактически сделано

Показано, что в результате перехода к асинхронным пересылкам общее время решения уравнений Максвелла снизилось на 20 %, позволяет сделать **вывод об эффективности реализованных асинхронных пересылок при использовании декомпозиции области**.

### Подробности реализации

На рисунке 7 показан этап работы параллельной программы по моделированию взаимодействия электронных пучков с плазмой, 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. 7.** 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 4.** 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 5.** 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 4 и 5 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**.

## Литература

1. B.M. Glinskiy, I.M. Kulikov, I.G. Chernykh, A.V. Snytnikov, A.V. Sapetina, D.V. Weins. The Integrated Approach to Solving Large-Size Physical Problems on Supercomputers. Принято к публикации в Communications in Computer and Information Science, изд-во Springer, 2017 г.
2. М.А.Боронина, А.В.Снытников. Разработка высокомасштабируемого параллельного алгоритма для моделирования динамики плазмы. //тезисы Международной конференции "Вычислительная и прикладная математика 2017".
3. A.V.Snytnikov.  Porting a Plasma Simulation PIC code from GPU cluster to a cluster built with Intel Xeon Phi accelerators. //в печати - Parallel Computing.