УДК 519.684

**А.А.Романенко, А.В.Снытников, И.Г.Черных**

*Новосибирский Государственный Университет  
630090, г.Новосибирск, ул.Пирогова, 2*

*Институт Вычислительной Математики и Математической Геофизики СО РАН,*

*630090, г.Новосибирск, пр. Лавретьева, 6*

[*arom@ccfit.nsu.ru*](mailto:arom@ccfit.nsu.ru)*,* [*snytav@ssd.sscc.ru*](mailto:snytav@ssd.sscc.ru)*, chernykh@parbz.sscc.ru*

**АДАПТАЦИЯ ПАРАЛЛЕЛЬНОГО ВЫЧИСЛИТЕЛЬНОГО АЛГОРИТМА К АРХИТЕКТУРЕ СУПЕРЭВМ НА ПРИМЕРЕ МОДЕЛИРОВАНИЯ ДИНАМИКИ ПЛАЗМЫ МЕТОДОМ ЧАСТИЦ В ЯЧЕЙКАХ[[1]](#footnote-2)**

Аннотация

Для повышения эффективности использования суперЭВМ и для ускорения проведения расчетов по моделированию турбулентной плазмы в трехмерной постановке существует необходимость динамической, непосредственно в процессе счета, адаптации вычислительного алгоритма к архитектуре суперЭВМ. Проведено большое количество физических расчетов с использованием программы в трехмерной расчетной области, использующей двухступенчатую эйлерово-лагранжеву декомпозицию расчетной области. Для адаптации программы к реально выделенным для ее работы вычислительным узлам в данной работе предложены два подхода. Во-первых, на основании тестовых расчетов на небольшом количестве процессорных ядер измерить время коллективных и парных коммуникаций MPI, построить аппроксимацию времени пересылок для произвольного количества процессов и выбрать оптимальное сочетание эйлеровой и лагранжевой декомпозиции. Во-вторых, необходимо провести анализ коммуникационной структуры выделенных для счета узлов суперЭВМ и перенумеровать процессы с учетом их расположения на узлах.

Ключевые слова: трехмерная модель, эйлерово-лагранжева декомпозиция, анализ коммуникационной структуры, оценка производительности.

## Введение

Актуальность настоящей работы связана с необходимостью проведения моделирования генерации высокочастотного (терагерцового) излучения, наблюдаемого в экспериментах [1-3] на многопробочной магнитной ловушке ГОЛ-3 (ИЯФ СО РАН) в трехмерной постановке. Моделирование задачи в двумерной постановке описано в [4-6]. Стоит отметить, что в трехмерной постановке задачи требуется проводить на порядки больше вычислений (количество узлов по третьей, координате, по Z должно быть более 100 в полномасштабной модели) и, таким образом существенно растут затраты по памяти (в данном случае на 2 порядка). Создание трехмерной программы было описано в [7,8].

С точки зрения программирования актуальность работы связана с разработкой параллельного высокопроизводительного комплекса программ, использующего несколько уровней параллелизма [6] (распараллеливание по узлам суперЭВМ, по отдельным процессорным элементам внутри узла и по отдельным ядрам процессора или ускорителя вычислений), а также с реализацией распределения загразки процессоров, проводящих вычисления с частицами, передвигающимися по области [9,10,11]. При этой достижение высокой вычислительной производительности является принципиально важным для получения физически значимых результатов.

## Математическая модель

Физическая система, состоящая из плазмы и электронного пучка, описывается системой уравнений, состоящей из уравнения Власова (отдельно для ионной и электронной компонент плазмы) и уравнений Максвелла [3]:

Здесь f – функция распределения (фазовая плотность), **r**,**v**, и **p** – координата, скорость и импульс, **E** и **B** – электрическое и магнитное поля, **j** – ток,  – плотность заряда. Эти уравнения решаются в безразмерной форме. Для перехода к безразмерным величинам используются следующие параметры:

* Скорость света c = 3×1010 см/с
* плотность плазмы n0 = 1014 см-3
* плазменная электронная частота *ωpe* = 1.7×1011 c-1

Граничные условия периодические по Y и Z, по X граничные условия непериодические для обеспечения вход и выход частиц пуска без создания нефизического потенциального барьера на границе.

## Численные методы

Уравнение Власова решается методом частиц в ячейках [3, 4,12,13]. В рамках этого метода решаются уравнения движения модельных частиц, которые представляют собой уравнения характеристик для уравнения Власова:

**,**

**,**где ***p*** – импульс частицы, ***E*** и ***B*** – электрическое и магнитное поля в точке, где находится частица, ***r*** – координата частицы, ***v*** – скорость частицы,  - отношение заряда частицы к массе,  – релятивистский гамма-фактор (жирным шрифтом показаны векторные величины).

## Моделирование прохождения пучка сквозь плазму

В данной работе рассматривается вариант непрерывного прохождения частиц пучка сквозь плазму. Поскольку реальные размеры установки (установка ГОЛ-3, ИЯФ СО РАН) [1] в рамках кинетической модели рассмотреть не представляется возможным, моделируется небольшой участок плазмы с частицами пучка. Для этого необходимо, чтобы частицы пучка входили в пространство моделирования и выходили из него.   
В программе на основе метода частиц это выполняется следующим образом. В левой части пространства моделирования

*0 < x < Lx, Ly1 < y < Ly2, Lz1 < z < Lz2*

где *Lx, Ly1 ,Ly2, Lz1 ,Lz2*–границы буфера**,** при этом Lx подбирается экспериментально, но не более 10% размера пространства моделирования. Размер буфера по Y и по Z выбирается из физических соображений (соотношение ширины пучка и плазмы).

В начальный момент времени частицы пучка распределены внутри буфера равномерно, также, как и электроны и ионы плазмы, так что в целом плотность заряда равна нулю, как и ток в целом по области [7]. В дальнейшем с течением времени частицы пучка вылетают за пределы буфера, и для того, чтобы обеспечить непрерывное прохождение пучка сквозь плазму, в буфер добавляются новые частицы пучка, с распределением, аналогичным начальному. Так как на частицы пучка внутри буфера поле не действует, то распределение импульсов не меняется со временем.

**Параллельная реализация**

Распараллеливание выполнено методом декомпозиции расчетной области по координате Y, т.е. по направлению, перпендикулярному направлению движения электронного пучка (пучок летит вдоль X), как показано на (рис.1).

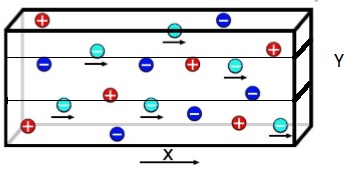


Рис.1. (**ris1.jpg**) Декомпозиция расчетной области (область делится по координате Y). Схематически показаны электроны пучка (голубым цветом), которые на начальном этапе движутся строго вдоль X, а также элетроны и ионы плазмы, направление движения которых произвольно.

Используется смешанная эйлерово-лагранжева декомпозиция [7,8,9]. Сетка, на которой решаются уравнения Максвелла, разделена на одинаковые подобласти по одной из координат. С каждой подобластью связана группа процессоров (в том случае, когда вычисления производятся на многоядерных процессорах, процессором для единообразия будет именоваться отдельное ядро). Далее, модельные частицы каждой из подобластей разделяются между процессорами связанной с этой подобластью группы равномерно, вне зависимости от координаты.

С этим связан основные вопросы настоящей статьи:

* что, если группа процессов, которые должны осуществлять коллективные пересылки, окажется размещенной на узлах суперЭВМ, расположенных физически далеко друг от друга, так что время выполнения этих пересылок будет велико?
* Как можно избежать такой ситуации, как объединять в группы для коллективных пересылок близко расположенные процессы, при том что MPI (или система очередей) размещает процессы на узлах фактически случайным образом?
* Как можно заранее оценить трудоемкость коммуникационных операций и выработать оптимальную схему размещения процессов?
* На основании чего можно принимать решение о перемещении процессов с одного узла на другой, или о перенумерации процессов?

В некоторых случаях эти вопросы решаются специально созданными внешними инструментами [10, 11], но для рассматриваемой в данной работе программы целесообразным является иметь собственные средства для решения проблемы неудачного размещения на узлах.

Для того, чтобы частично ответить на эти вопросы, были проведены вычислительные эксперименты на нескольких суперЭВМ с измерением производительности различных коммуникационных операций, построены оценки трудоемкости этих операций и предложен метод выделения групп близко расположенных процессов. Создать правильное представление о реальной длительности коммуникаций между процессами в программе важно также для того, чтобы решить вопрос о целесообразности использования динамической балансировки и для выбора конкретного ее варианта [12-14], и о применении форм-факторов высокого порядка [15-16] в методе частиц в ячейках [17,18, 19].

## Краткое описание вычислительных экспериментов

Задавались следующие основные параметры и числовые характеристики тестовых расчетов:

* *NX, NY, NZ,* - размер сетки по каждому из измерений (*NX, NY –* от 100 до 500, NZ = 20)
* PALL - общее количество процессорных ядер (до 1000)
* PSUB - число подобластей (до 20)

Измерялись кроме физических величин, следующие времена (с помощью Intel Trace Analyzer &Collector, https://software.intel.com/en-us/intel-trace-analyzer):

* 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.

Расчеты проводились на следующих суперЭВМ:

* Кластер НГУ. Из различных имеющихся типов узлов использовались только узлы  HP BL2x220c G6, каждый из которых содержит две материнские платы, на каждой из которых: два 4-ядерных процессора Intel Xeon E5540 с тактовой частотой 2530 МГц и 16 ГБ ОЗУ
* Кластер СПбПУ «Политехник». Использовались узлы с двумя 28-ядерными процессорами  Intel®Xeon® E5-2600 v3
* Кластер «Ломоносов» в НИВЦ МГУ. Использовались узлы основного раздела, содержащие 2 4-ядерных процессора Intel Xeon  X5570

Физические параметры проводимых в данной работе расчетов соответствуют кинетическому режиму развития двухпотоковой неустойчивости, рассмотренному в [20].

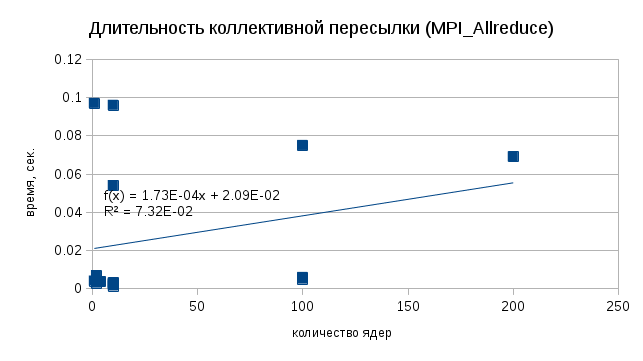
## Верификация формул для оценки времени работы коммуникационных процедур

Для оценки времени работы коммуникационных процедур, в первую очередь MPI\_Allreduce и MPI\_Send/MPI\_Recv с целью выработки оптимальной схемы размещения процессов исходя из реально выделенных (системой очередей) вычислительных ресурсов, можно использовать следующие простые формулы:

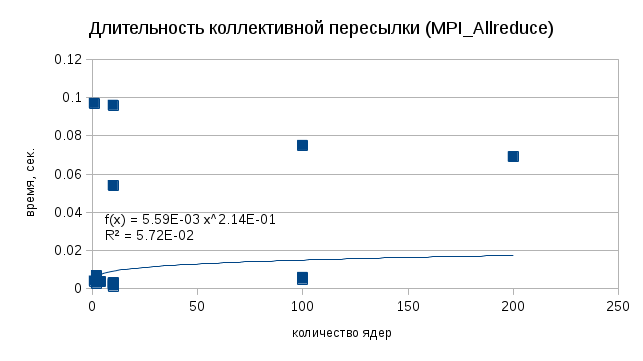
,

где T- время пересылки данных, N – количество процессов (ядер), a и b – константы, зависящие от архитектуры суперЭВМ, количества пересылаемых данных, реализации MPI и др. На рисунках 2-4 и 5-7 показаны результаты аппроксимации реальных данных о продолжительности пересылок, полученных в ходе вычислительных экспериментов, описанных в предыдущем разделе. На рисунках приведены аппроксимирующие формулы, и указано значение среднеквадратической ошибки, позволяющее определить наилучший тип аппроксимации. На рисунках 2-4 и 5-7 показаны данные, измеренные на всех кластерах, на которых проводились расчеты, т.е. особенности архитектуры здесь не учитываются.

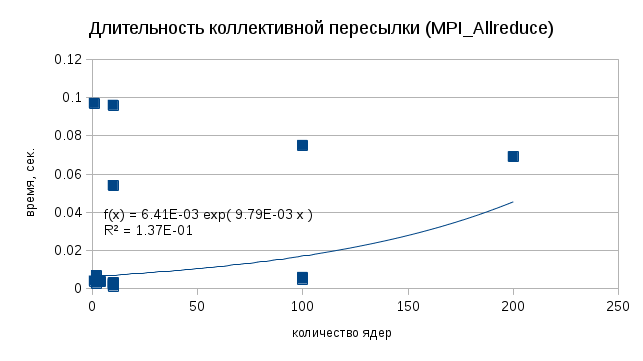
Каждая точка на рисунках 2-4 и 5-7 представляет собой отдельный расчет. Все расчеты проведены на разных архитектурах, количество ядер соответствует количеству MPI-процессов. Возможно, что запуски на 50, 150 ядер аннулируют все догадки и предположения, а может наоборот вычертят кривую, т.е. полученные результаты имеют предварительный характер.



**Рис.2.** Продолжительность операции MPI\_Allreduce. Линейная аппроксимация.



**Рис.3.** Продолжительность операции MPI\_Allreduce. Степенная аппроксимация.

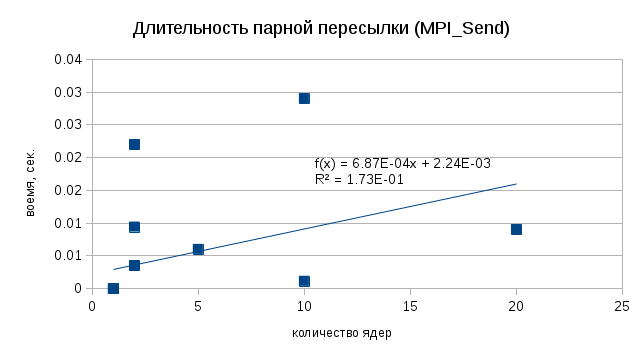


**Рис.4.** Продолжительность операции MPI\_Allreduce. Экспоненциальная аппроксимация.

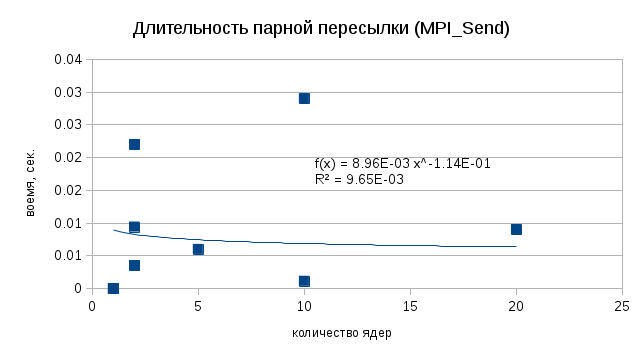
Из рисунков 2-4 видно, что наименьшее значение среднеквадратической ошибки достигнуто при степенной аппроксимации:

*T = 0.0056\*N0.214*

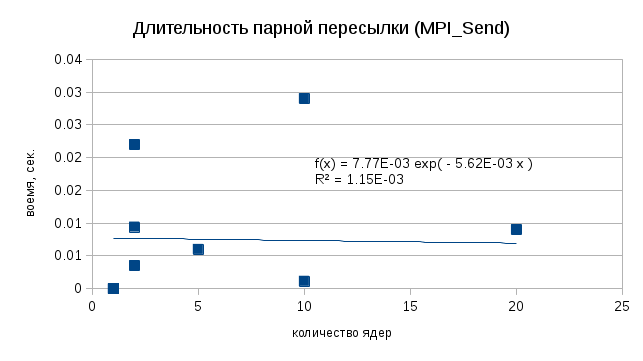
Далее рассмотрим парные пересылки. Зависимость от числа ядер в данном случае возникает потому, что в таких пересылках задействованы все ядра, между которыми разделена область: MPI-процесс, работающий на каждом ядре пересылает данные обоим своим соседям. Фактически речь идет о длительности эйлерова этапа параллельного алгоритма.



**Рис.5.** Продолжительность операции MPI\_Send. Линейная аппроксимация.



**Рис.6.** Продолжительность операции MPI\_Send. Степенная аппроксимация.



**Рис.7.** Продолжительность операции MPI\_Send. Экспоненциальная аппроксимация.

В этом случае наименьшая ошибка оказалась достигнута при экспоненциальной аппроксимации:

*T = 0.0077\*exp(-0.00562\*N)*

Полученные формулы не является зависимостями, работающими всегда, на любых суперЭВМ, любых реализациях MPI и т.д. Их назначение в том, чтобы в ходе реального крупномасштабного расчета, без повторных запусков, не прерывая счет, ответить на вопрос, что будет, если увеличить количество процессоров (ядер), вовлеченных в коллективные взаимодействия. Т.е. в данном случае предполагается динамическое дозапускание MPI процессов, допустимое стандартом MPI2.

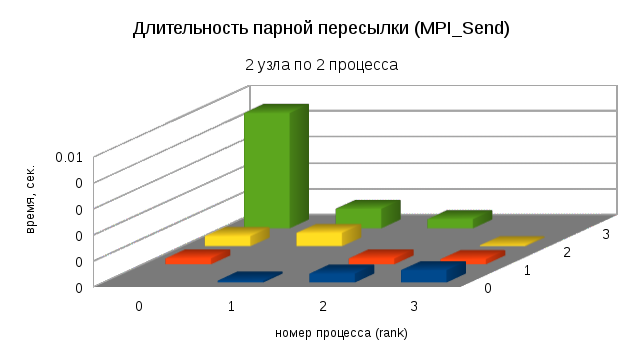
Например, программа, использующая 40 MPI-процессов получила для счета 40 ядер. Эти ядра можно по-разному распределить между эйлерой и лагранжевой декомпозицией области: можно поделить область на 10 частей, и затем распределить все частицы каждой подобласти между 4 MPI-процессами, а можно наоборот. Даже для 40 процессов есть несколько вариантов, в то время как речь идет о расчетах на нескольких тысячах ядер, где невозможно будет просто перебрать все варианты и выбрать оптимальный.

Для того, чтобы сделать правильный выбор, предлагается после выделения узлов для счета провести несколько тестовых запусков коллективных операций, построить для данного конкретного расчета аппроксимацию, подобную полученной выше, и на ее основе принимать решение.

## Выделение групп близко расположенных процессов.

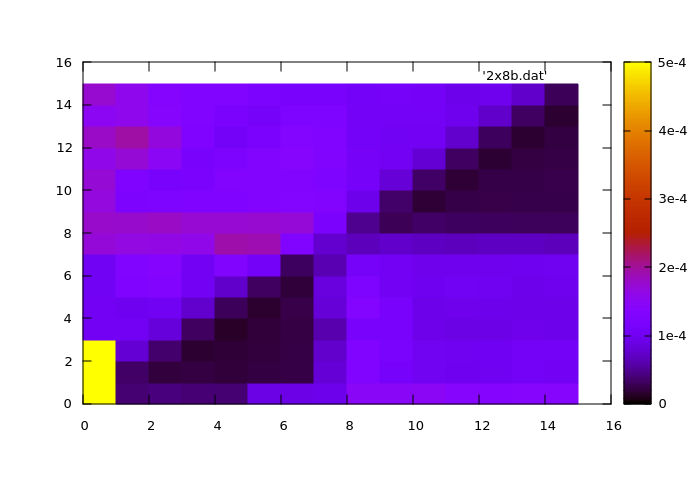
MPI предоставляет пользователю возможность создания собственных виртуальных топологий, в том числе декартовых (двумерных, трехмерных и пр.). При этом в соответствии со стандартом процессы, расположенные на физически близких узлах, должны иметь близкие номера в рамках топологии, однако все зависит от конкретной реализации MPI.

Для решения этого вопроса в описанной программе реализован специальный диагностический модуль, выполняющий пересылки типа «точка-точка» (MPI\_Send/MPI\_Recv) между всеми процессами (all-to-all, «каждый с каждым»). При этом рассматривались разные варианты размещения процессов по узлам. В каждом случае измерялось время пересылки с помощью функции MPI\_Wtime. На рисунках 8-13 показано время пересылок во всех парах взаимодействующих процессов.



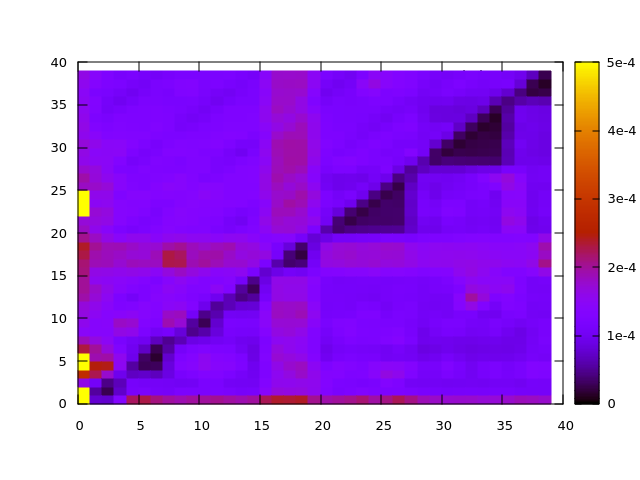
**Рис.8.** Время пересылок для 4 MPI-процессов, расположенных на двух узлах попарно, кластер НГУ.

На рисунке 2 видно, что время пересылок внутри узла (0-й процесс и 1-й, или 3-й и 4-й) меньше, чем время пересылок между узлами. Для того, чтобы аналогичным образом рассмотреть более сложные конфигурации, необходимо перейти от трехмерной столбчатой диаграммы к двумерным картам плотности, например, рис.9.



**Рис.9.** Время пересылок для 16 MPI-процессов, расположенных на двух узлах по 8 процессов, кластер НГУ. По осям X и Y отложены номера процессов, цветовая шкала показывает время пересылок в секундах.

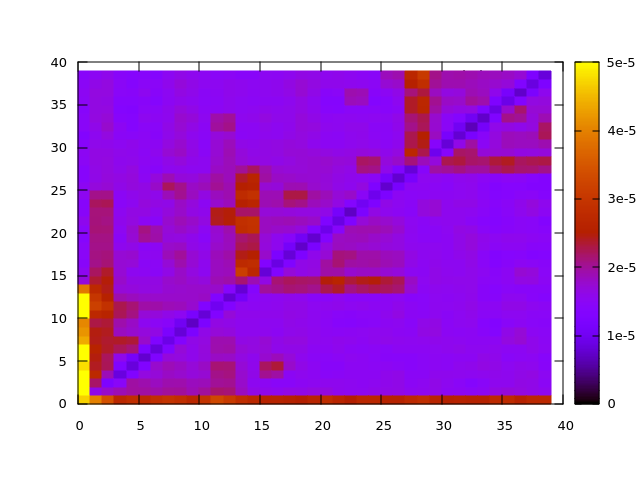
Рисунок 9 естественным образом разбивается на 4 зоны, соответствующих размещению процессов по узлам, несмотря на то, что принципиальной разницы по времени пересылок между узлами и внутри узлов в данном случае нет.



**Рис.10.** Время пересылок для 40 MPI-процессов, расположенных на 10 узлах по 4 процесса, кластер НГУ. По осям X и Y отложены номера процессов, цветовая шкала показывает время пересылок в секундах.

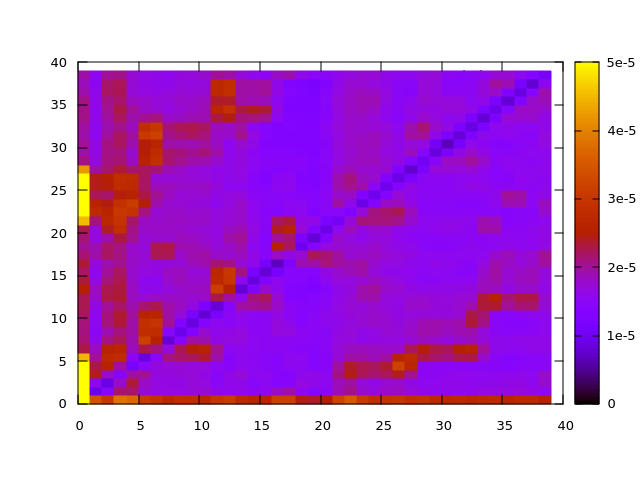
На рисунке 10 не удается зрительно выделить 10 зон, соответствующих 10 узлам, на которые проводился расчет. Тем не менее видно, что участки с наименьшими временами пересылок расположены вдоль диагонали матрицы, как и на предыдущих рисунках.

Следует отметить, что такая задача безусловно и не должна решаться «на глаз», в дальнейшем планируется применить здесь известные методы выделения сообществ в полном графе. Задача данной работы только в том, чтобы проверить обмен сообщениями у попарно между всеми процессами как метод тестирования архитектуры кластера: позволяет ли он выявлять отличие между близко и далеко расположенными процессами. Для этого обратимся к суперЭВМ существенно отличной архитектуры, а именно кластеру «Политехник» в СПбПУ.



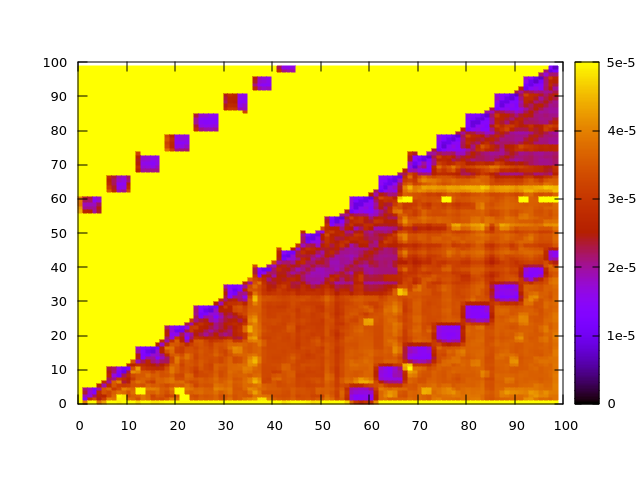
**Рис.11.** Время пересылок для 40 MPI-процессов, расположенных на 10 узлах по 4 процесса, кластер «Политехник», СПбПУ. По осям X и Y отложены номера процессов, цветовая шкала показывает время пересылок в секундах.

В первую очередь важно отметить, что на рисунке 11 все времена на порядок меньше, чем на рис. 10, показывающем ту же конфигурацию для кластера НГУ. Далее, на рис.11 видны прямоугольные участки с большим временем пересылок (красного цвета), шириной в 4 процесса, расположенные как горизонтально, так и вертикально. Это говорит о том, что возможна ситуация, когда близко по номеру расположенные процессы будут иметь большее время обмена сообщениями по сравнению с более удаленными (в данном случае 13-15 процессы при обмене с 16-20 процессами).



**Рис.12.** Время пересылок для 40 MPI-процессов, расположенных на 4 узлах по 10 процессов, кластер «Политехник», СПбПУ. По осям X и Y отложены номера процессов, цветовая шкала показывает время пересылок в секундах.

На рис. 12 показан расчет также с использованием 40 процессов, но размещенных на 4 узлах. Видно, что красные зоны (с большим временем пересылок) имеют больший размер и преимущественно локализованы в левой верхней и правой нижней четвертях квадрата, что соответствует обмену сообщениями между процессами, расположенными на разных узлах.



**Рис.13.** Время пересылок для 100 MPI-процессов, расположенных на 10 узлах по 10 процессов, кластер «Политехник», СПбПУ. По осям X и Y отложены номера процессов, цветовая шкала показывает время пересылок в секундах.

На рисунке 13 видны группы размером 5 процессов, время обмена между которыми заметно меньше, чем со всеми остальными. Это также коррелирует с их размещением по узлам.

По итогам анализа рисунков 2-7 можно сделать вывод, что использованный метод анализа архитектуры суперЭВМ позволяет обнаружить группы близко расположенных процессов.

**Заключение**

В заключение можно дать следующие рекомендации для использования конкретного варианта декомпозиции в зависимости от реально доступной коммуникационной структуры суперЭВМ:

* Использовать для лагранжевой декомпозиции только близко (внутри одного узла, или на соседних узлах) расположенные процессы, или в рамках узла использовать OpenMP.
* Не назначать соседние подобласти в рамках эйлеровой декомпозиции на далеко разнесенные узлы
* Приоритетной считать эйлерову декомпозицию в том случае, если память узла позволяет разместить соответствующий фрагмент данных

## Литература

1. V.T. Astrelin, A.V. Burdakov, V.V. Postupaev, Generation of ion-acoustic waves and suppresion of heat transport during plasma heating by an electron beam. //Plasma Physics Reports. 24(5) 414–425, 1998.
2. Терехов А.В., Тимофеев И.В., Лотов К.В. Двумерная численная модель плазмы для изучения процессов плазменно-пучкового взаимодействия.  Вестник Новосибирского государственного университета. Серия: Физика. 5(2):85-97. 2010.
3. A. V. Burdakov, I. A. Kotelnikov, V. I. Erofeev. [Explanation of Turbulent Suppression of Electron Heat Transfer in GOL-3 Facility at the Stage of Relativistic Electron Beam Injection](http://www.tandfonline.com/doi/full/10.13182/FST05-A611). [Fusion Science and Technology](http://www.tandfonline.com/toc/ufst20/47/1T)Vol. 47 , Iss. 1T, 2005.
4. V. V. Annenkov, I. V. Timofeev, and E. P. Volchok. Simulations of electromagnetic emissions produced in a thin plasma by a continuously injected electron beam. //Physics of Plasmas 23, 053101, 2016.
5. I. V. Timofeev, E. P. Volchok, and V. V. Annenkov. Physics of Plasmas 2016 23:8.
6. I. V. Timofeev, E. A. Berendeev, and G. I. Dudnikova. Physics of Plasmas 2017 24:9.
7. А.А.Романенко, А.В.Снытников, И.В.Тимофеев. Трехмерный гибридный код для моделирования генерации высокочастотного электромагнитного излучения турбулентной плазмы. Вестник Новосибирского государственного университета. Серия: Информационные технологии.. 14(3):81-90. 2016.
8. Aleksey Snytnikov, Supercomputer simulation of plasma electron heat conductivity decrease due to relativistic electron beam relaxation, In Procedia Computer Science, Volume 1, Issue 1, 2010, Pages 607-615, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2010.04.065.
9. Ф.О.Голомидов, А.А.Воропинов, И.Г.Новиков. Развитие OpenMP распараллеливания в методике ТИМ. «Супервычисления и математическое моделирование». Труды XVI Международной конференции. Саров, 2016. стр. 100-102.
10. Городничев М.А.   [Объединение вычислительных кластеров для крупномасштабного численного моделирования в проекте NumGRID](http://ssd.sscc.ru/ru/publications/объединение-вычислительных-кластеров-для-крупномасштабного-численного-моделирования-в)**.**Вестник НГУ. Серия: Информационные технологии. 10(4):10. 2012.
11. Малышкин В.Э., Арыков С.Б.   [Алгоритмы конструирования асинхронных программ заданной степени непроцедурности методом группировки](http://ssd.sscc.ru/ru/node/249). Вестник Новосибирского государственного университета. Серия: Информационные технологии.. 7(1):3-15. 2009.
12. C. Othmer, J. Schüle, Dynamic load balancing of plasma particle-in-cell simulations: The taskfarm alternative, In Computer Physics Communications, Volume 147, Issues 1–2, 2002, Pages 741-744, ISSN 0010-4655, <https://doi.org/10.1016/S0010-4655(02)00389-2>.
13. Roel Wuyts, Tom Haber, Giovanni Lapenta, Helsim: A Particle-in-cell Simulator for Highly Imbalanced Particle Distributions, In Procedia Computer Science, Volume 51, 2015, Pages 2923-2927, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2015.05.479>.
14. M.A. Kraeva, V.E. Malyshkin, Assembly technology for parallel realization of numerical models on MIMD-multicomputers, In Future Generation Computer Systems, Volume 17, Issue 6, 2001, Pages 755-765, ISSN 0167-739X, <https://doi.org/10.1016/S0167-739X(00)00058-3>.
15. Nicolas Crouseilles, Pierre Navaro, Éric Sonnendrücker, Charge-conserving grid based methods for the Vlasov–Maxwell equations, In Comptes Rendus Mécanique, Volume 342, Issues 10–11, 2014, Pages 636-646, ISSN 1631-0721, <https://doi.org/10.1016/j.crme.2014.06.012>.
16. T.D. Pointon, Second-order, exact charge conservation for electromagnetic particle-in-cell simulation in complex geometry, In Computer Physics Communications, Volume 179, Issue 8, 2008, Pages 535-544, ISSN 0010-4655, https://doi.org/10.1016/j.cpc.2008.04.017.
17. Ч. Бэдсел, А. Лэнгдон. «Физика плазмы и численное моделирование», Москва, Энергоатомиздат, 1989.
18. Р. Хокни, Дж. Иствуд. Численное моделирование методом частиц. 1987
19. Григорьев Ю.Н., Вшивков В.А., Федорук М.П. Численное моделирование методами частиц в ячейках. Новосибирск, Изд-во СО РАН, 2004.
20. Е.А. Месяц, А.В. Снытников, К.В. Лотов. О выборе числа частиц в методе частиц-в-ячейках для моделирования задач физики плазмы. //Вычислительные Технологии, стр. 83-96. Том 18, № 6, 2013.

1. Работа выполнена при поддержке гранта РФФИ 16-07-00434. Вычислительные эксперименты на суперЭВМ частично поддержаны грантом РФФИ 16-01-00209. [↑](#footnote-ref-2)