

# ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ МНОГОЯДЕРНЫХ СОПРОЦЕССОРОВ ПРИ СУПЕРКОМПЬЮТЕРНОМ СТАТИСТИЧЕСКОМ МОДЕЛИРОВАНИИ ЭЛЕКТРОННЫХ ЛАВИН

М.А. Марченко

## Аннотация

Для моделирования развития электронных лавин в газе разработаны трехмерный параллельный алгоритм метода Монте-Карло и программа ELSHOW, реализованная с использованием комбинирования принципов крупно- и мелкозернистого параллелизма. Для реализации параллельных вычислений на высокопроизводительных гибридных вычислительных системах с сопроцессорами Intel Phi используется хорошо зарекомендовавшая себя библиотека PARMONC. Применение разработанной технологии распараллеливания существенно уменьшает вычислительную трудоемкость оценки таких интегральных характеристик, как число частиц в лавине, коэффициент ударной ионизации, скорость дрейфа и др.

## Введение

Во многих приложениях физики импульсного пробоя газов возникает необходимость изучения динамики процесса развития электронной лавины во времени, в частности, описание предпробойной стадии для условий пробоя сильно перенапряженных промежутков [1]. Известно, что метод Монте-Карло успешно применяется для решения задач переноса заряженных частиц в веществе [2, 3].

Основной проблемой использования метода Монте-Карло для моделирования ионизационного размножения электронов является лавинообразное нарастание количества электронов и ионов (моделируемых частиц, для каждой из которых необходимо решать уравнения движения). Для решения этой проблемы можно использовать различные методы укрупнения (например, когда моделируемая частица представляется в виде облака, содержащего в себе несколько элементарных частиц), что приводит к ухудшению точности моделирования. Целесообразно, однако, учитывать траектории каждого отдельно взятого электрона и использовать для моделирования эффективные технологии распараллеливания на многопроцессорных высокопроизводительных вычислительных системах [2, 4].

Решаемая в настоящей работе задача возникла в рамках выполнения междисциплинарного интеграционного проекта СО РАН №126 "Импульсные разряды в газах высокого давления и моделирование газоразрядных, плазмохимических и газодинамических процессов методом Монте-Карло" в ИВМиМГ СО РАН, где работает автор, и в ИСЭ СО РАН, г. Томск. Координатор проекта - акад. Б.М. Ковальчук (ИСЭ СО РАН). Коллектив соисполнителей в ИВМиМГ СО РАН: чл.-корр. Г.А. Михайлов, д.ф.-м.н. С.В. Рогазинский (координаторы проекта в ИВМиМГ СО РАН), д.ф.-м.н. С.А. Ухинов, к.ф.-м.н. Г.З. Лотова, к.ф.-м.н. М.А. Марченко. Результаты автора относятся, в частности, к параллельной реализации трудоемких вычислений, которые необходимо осуществлять в процессе моделирования электронных лавин.

## Вероятностная модель развития электронных лавин

Отличительной особенностью метода Монте-Карло, используемого для расчета ионизационных лавин, по сравнению с другими методами (например, методом расчета функции распределения или численного решения уравнений переноса) является возможность изучения динамики процесса развития лавины во времени и влияния на нее незначительных флуктуаций. Основной задачей моделирования развития ионизационных лавин является описание предпробойной стадии (т.е., до момента, когда собственное электрическое поле электронов и ионов лавины становится сравнимым с внешним электрическим полем) для условий пробоя сильно перенапряженных промежутков. Решение этой задачи основано на использовании метода Монте-Карло в модели индивидуальных столкновений, в которой вероятность столкновения определяется сечением процесса или процессов. Подробно предлагаемая вероятностная модель описана в [4].

Упомянутым коллективом авторов создана программа ELSHOW (ELectron SHOWer). Программа реализует параллельный трехмерный алгоритм статистического моделирования, который учитывает ускорение электронов в электрическом поле, процессы упругого рассеяния электронов на молекулах газа и ионизации, а также возбуждения. Для этого используются соответствующие сечения 24-х типов взаимодействий для азота.

Рассматривается открытая система с внешним электрическим полем, напряженность которого считается постоянной. В ходе моделирования с катода из начальной точки в нулевой момент времени происходит выброс некоторого числа электронов с нулевыми энергиями. Затем прослеживаются траектории движения каждого из электронов, а также всех вторичных электронов, образовавшихся в результате ионизации, до достижения максимального физического времени  $t_{max}$ , которое соответствует моменту предпробойного состояния. С этой целью делаются одинаковые малые шаги по времени такие, чтобы при этом соответствующий пробег был меньше заданной доли длины свободного пробега для любых энергий. В конце каждого шага разыгрывается столкновение с вероятностью, зависящей от полного микроскопического сечения

взаимодействия, начальной энергии электрона, концентрации частиц газа. Затем разыгрывается тип взаимодействия в соответствии с сечениями упругого рассеяния, возбуждения и ионизации.

Упругое рассеяние не меняет энергию частицы, а направление движения определяется согласно известным дифференциальным сечениям [1, 2]. При этом азимутальный угол выбирается равновероятно на отрезке от нуля до 360 градусов. Для всех видов возбуждения энергия уменьшается на величину пороговой энергии, а направление движения разыгрывается так же, как и для упругого рассеяния. При ионизации энергия налетающего электрона сначала уменьшается на величину пороговой энергии ионизации, а затем остаток делится определенным образом между двумя вылетающими электронами [4]. Из простых кинематических уравнений можно получить направления движения образующихся в результате ионизации электронов.

В программе ELSHOW моделирование индивидуальных соударений построено таким образом, что прослеживается путь частицы от момента вылета с катода до момента времени  $t_{max}$ . При этом в массив фактически записываются лишь вторичные частицы, которые образовались в результате ионизации. Затем моделируется путь одной из вторичных частиц до достижения момента времени  $t_{max}$ , и так далее до тех пор, пока все вторичные частицы не будут рассмотрены. Это называется лексикографической схемой. Она не требует больших объемов памяти для хранения всех частиц. Кроме того, заложена возможность использования метода укрупненных соударений, в котором на определенном шаге по времени производится процедура удаления электронов с заданной вероятностью (т.н. «русская рулетка»). Для оставшихся частиц статистический «вес» умножается на соответствующий вероятности множитель. Если при этом число частиц достаточно велико, то энергетический спектр сохраняется.

Программа ELSHOW в результате своей работы через равные промежутки времени выдает значения числа частиц, положения центра масс, средней скорости, средней энергии, гистограммы плотности частиц, а также их среднеквадратические погрешности. Это позволяет вычислить различные характеристики лавины такие, как скорость центра масс и скорость дрейфа, коэффициенты поперечной и продольной диффузии, коэффициент ударной ионизации [1, 3, 4]. Более того, все числовые характеристики можно получить и для нескольких промежуточных значений времени, что дает возможность исследовать их динамику.

Сравнение поведения характерного функционала, полученного в наших расчетах, с результатами экспериментов приведено на рис. 1. На рис. 2 представлено распределение мест ионизации (ионный хвост лавины) в моделировании. Проведенные расчеты подтверждают адекватность выбранной вероятностной модели развития электронных лавин в газе.

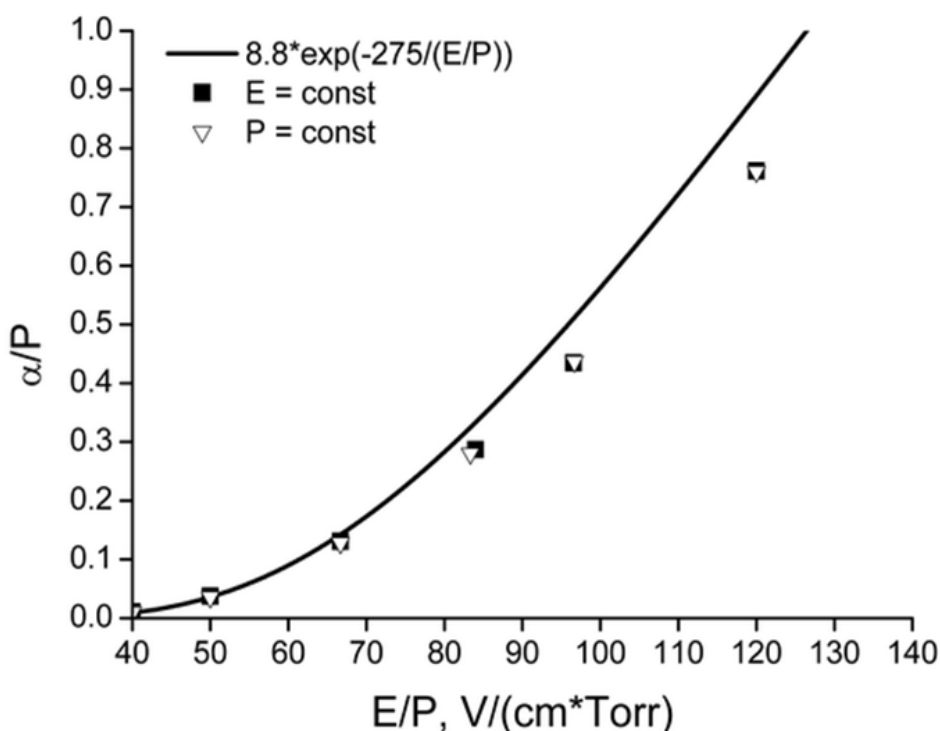


Рис. 1. Приведенный коэффициент ударной ионизации для азота. Сплошная кривая - аппроксимация из экспериментальных данных, точки - моделирование методом Монте-Карло

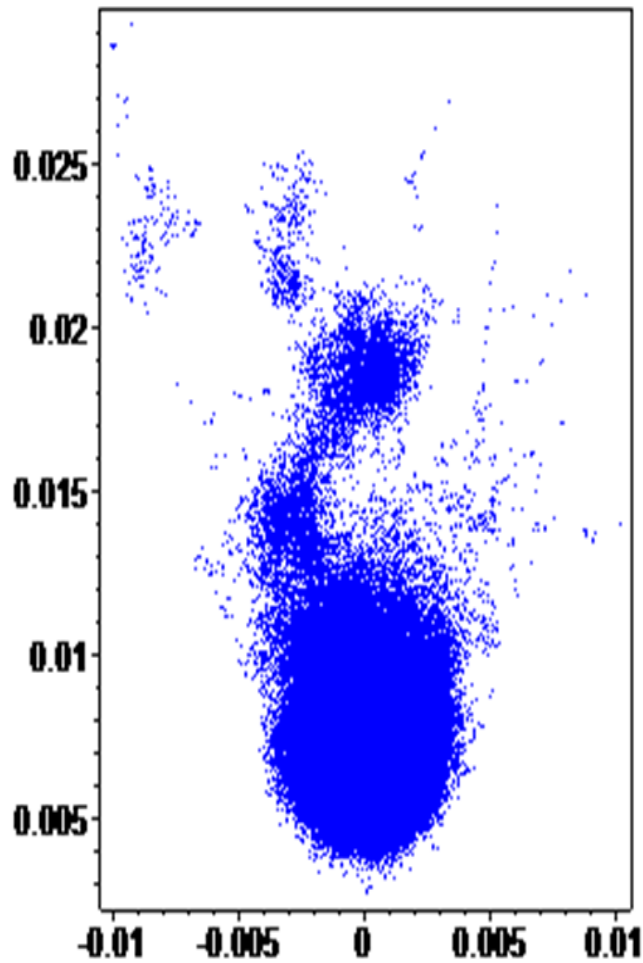


Рис. 2. Распределение мест ионизации (ионный хвост лавины) - сечение в плоскости, перпендикулярной катоду. Физическое время - 0.05 нс

### Технологии распараллеливания статистического моделирования

Вероятностные модели, описывающие развитие электронных лавин, обладают большой вычислительной сложностью. Тем не менее, такого рода модели могут быть успешно реализованы с использованием распараллеливания на многопроцессорных вычислительных системах, например, с графическими ускорителями [5]. В программе ELSHOW распараллеливание осуществлялось в рамках единого подхода к реализации распределенного статистического моделирования, предложенного в работе [6]. Применяемая методология позволяет добиться высокой эффективности в использовании гибридных вычислительных систем и существенного уменьшения трудоемкости вычислений: в наших расчетах моделирование развития лавин электронов проводилось вплоть до 109 частиц без ухудшения точности оценки функционалов.

1. Метод *распределенного статистического моделирования* состоит в распределении моделирования независимых реализаций по вычислительным ядрам с периодическим усреднением полученных выборочных значений по статистически эффективной формуле

$$\bar{\zeta} = \left[ \sum_{m=1}^M l_m \right]^{-1} \sum_{m=1}^M l_m \bar{\zeta}_m \quad (1)$$

Здесь  $\bar{\zeta}$  - искомое выборочное среднее значение,  $M$  - общее число ядер,  $l_m$  - объем выборки, полученной на  $m$ -м ядре,  $\bar{\zeta}_m$  - соответствующее  $m$ -му ядру выборочное среднее значение.

Очевидно, что главным критерием осуществимости такой параллельной реализации является возможность «поместить» данные вычислительной программы для моделирования реализаций в оперативную память каждого ядра. Подчеркнем, что в целях распределенного статистического моделирования допустимо

использовать вычислительные ядра с разной производительностью. Важно, чтобы пересылка данных на центральное ядро и соответствующий прием данных осуществлялись в асинхронном режиме.

2. Как правило, при параллельной реализации необходимый объем выборки базовых случайных чисел очень велик, поэтому целесообразно использовать длиннопериодные псевдослучайные последовательности. А именно, для решения «больших» задач по методу Монте-Карло предлагается использовать генератор следующего вида:

$$u_0 = 1, u_n \equiv u_{n-1} A \pmod{2^{128}}, \alpha_n = u_n 2^{-128}, n = 1, 2, \dots$$

$$A \equiv 5^{100109} \pmod{2^{128}} \quad (2)$$

Получаемая последовательность псевдослучайных чисел является периодической, длина ее периода равна 2126.

Для реализации статистического моделирования на независимых вычислительных ядрах предлагается следующий *распределительный способ использования базовых псевдослучайных чисел* (термин предложен член-корр. Г.А. Михайловым). Базовая последовательность предварительно разбивается на подпоследовательности, начинающиеся с чисел  $\{\alpha_{m\mu}\}$ ,  $m = 0, 1, 2, \dots$ , где  $\mu$  -длина «прыжка», после чего полученные таким образом подпоследовательности распределяются по разным ядрам. Значение «прыжка» генератора должно выбираться так, чтобы такого количества псевдослучайных чисел хватало для моделирования на каждом ядре. Легко показать, что для метода вычетов начальные значения указанных подпоследовательностей получаются по формуле

$$u_{(m+1)\mu} = u_{m\mu} A_\mu \pmod{2^{128}}, \alpha_{m\mu} = u_{m\mu} 2^{-128}, m = 0, 1, \dots$$

$$A_\mu \equiv A^\mu \pmod{2^{128}} \quad (3)$$

Отметим, что длина периода генератора позволяет независимым образом распределять псевдослучайные числа по реализациям на практически неограниченное число вычислительных ядер. Параллельный генератор, приведенный выше, успешно используется в ряде институтов СО РАН на протяжении последних десяти лет.

3. С целью унификации использования распределенного статистического моделирования при решении широкого круга задач методом Монте-Карло автором разработана и внедрена программная библиотека **PARMONC** (**PAR**allel **MON**te **CAR**lo) [7]. Библиотека PARMONC установлена на кластерах Сибирского суперкомпьютерного центра (ЦКП ССКЦ СО РАН) и может использоваться на вычислительных системах с аналогичной архитектурой. Библиотека предназначена для распараллеливания программ, написанных на языках Fortran или C, причем библиотечные подпрограммы применяются без явного использования команд MPI. Инструкции по использованию библиотеки приведены в [8].

Возможность применения библиотеки PARMONC определяется «естественной» крупноблочной фрагментированностью программ статистического моделирования. При распараллеливании последовательных программ с помощью PARMONC определяется процедура *realization* (моделирующая подпрограмма), возвращающая одну реализацию (возвращение осуществляется через аргумент процедуры). При этом считается, что моделирующая подпрограмма использует потоки псевдослучайных чисел, генерируемых внешней по отношению к ней подпрограммой. Цикл по независимым реализациям и финальное осреднение заменяются вызовом библиотечной процедуры следующего вида:

```
parmoncc( realization, L, SUBT, ... );
```

Здесь имя моделирующей подпрограммы и общее число независимых реализаций передаются в библиотечную процедуру parmoncc в качестве входных аргументов; выборочное среднее будет возвращаться в переменную SUBT; для простоты остальные аргументы процедуры parmoncc опущены и заменены многоточием. Процедура parmoncc автоматически распределяет моделирование независимых реализаций по вычислительным ядрам. Все остальные операторы пользовательской программы остаются без изменений.

В процессе распределенных вычислений на каждом ядре используются потоки независимых псевдослучайных чисел, получаемые в результате работы подпрограммы, реализующей параллельный генератор. В процедуре *realization* библиотечный параллельный генератор вызывается следующим образом:

```
a = rnd128();
```

Здесь  $a$  — очередное псевдослучайное число, равномерно распределенное в интервале от нуля до единицы. Инициализация параллельного генератора выполняется автоматически при запуске программы, скомпилированной и собранной с помощью библиотеки PARMONC.

Следует упомянуть о практически важной возможности коррелирования результатов различных расчетов для одной и той же задачи, когда варьируется лишь ряд ее параметров. При использовании библиотеки PARMONC это делается на основе использования одних и тех же псевдослучайных чисел в каждом из расчетов. Предлагаемый выше параллельный генератор позволяет распределять псевдослучайные числа также и по отдельным траекториям в каждом из расчетов. С использованием такого подхода проводились многократные численные расчеты по подбору параметров нашей вероятностной модели, в частности, величины временного шага в модели.

4. Для рассматриваемой задачи *методика крупнозернистого распараллеливания* заключается в распределении моделирования отдельных реализаций электронной лавины по отдельным ядрам. С помощью библиотеки PARMONC такого рода распределение делается автоматически. Однако, с одной стороны, среднее время моделирования отдельной реализации может быть достаточно большим (в некоторых расчетах оно достигало нескольких суток!). С другой стороны, при крупнозернистом распараллеливании возникают проблемы, связанные с обработкой быстро растущего числа частиц в лавине.

Расчеты по методике крупнозернистого распараллеливания производились на кластере НКС-30Т в ЦКП ССКЦ СО РАН. Объем оперативной памяти, доступный каждому CPU ядру на узле, составляет от 4 до 8 Гб. Таких объемов оперативной памяти вполне достаточно для реализации методики распараллеливания. Количество используемых ядер варьировалось в пределах от 128 до 512.

*Методика мелкозернистого распараллеливания* для рассматриваемой задачи заключается в моделировании отдельной реализации электронной лавины на одном многоядерном процессоре, например, на графическом сопроцессоре или акселераторе Intel Phi. С этой целью при реализации лексикографической схемы моделирование развития условно-независимых «ветвей» дерева (т.е. части лавины частиц) от вторичных частиц, появляющихся при актах ионизации, распределяется по разным ядрам сопроцессора. Естественно, необходимо сбалансированно распределять моделирование «ветвей» по ядрам сопроцессора с целью недопущения простоя ядер, таким образом, осуществляя балансировку вычислительной нагрузки. Такая балансировка (т.е. пересылка массивов вторичных частиц), очевидно, требует затрат машинного времени.

При лавинообразном росте числа частиц в реализации лавины следует также передавать «лишние» частицы в память CPU ядра для временного размещения в «магазине» памяти и при окончании моделирования всех «ветвей» на сопроцессоре, догружать частицы в память сопроцессора из «магазина» для дальнейшей обработки.

Ясно, что при таком распараллеливании необходимо распределять псевдослучайные числа также и по отдельным ядрам сопроцессора. Применение методики мелкозернистого распараллеливания помогает обрабатывать быстро растущее число частиц в лавине.

Поскольку узлы гибридного кластера имеют в своем составе разные вычислители (CPU ядра и сопроцессоры), то целесообразно комбинировать крупно- и мелкозернистое распараллеливание следующим образом. На каждом вычислительном узле часть CPU ядер будет моделировать реализации согласно методике крупнозернистого распараллеливания, в то время как сопроцессоры узла (и «прикрепленные» к ним CPU ядра) будут моделировать реализации лавины по методике мелкозернистого распараллеливания. Таким образом, при использовании такой методики комбинированного распараллеливания вычислительные ресурсы гибридного узла будут задействованы максимально полно.

Расчеты по методике комбинированного распараллеливания производились на прототипе суперкомпьютера МВС-10П в МСЦ РАН. На этом кластере на каждом вычислительном узле доступны два 8-ми ядерных процессора Intel® Xeon® E5-2690 и два 60-ти ядерных сопроцессора Intel Xeon Phi SE10X. Объем оперативной памяти, доступный каждому CPU ядру на узле, составляет 4 Гб. На каждом сопроцессоре доступно 8 Гб оперативной памяти. Таких объемов оперативной памяти вполне достаточно для реализации описанных выше методик распараллеливания. При программной реализации использовалась технология т.н. offload-режима исполнения задач на сопроцессоре [9].

Для одной из физических постановок задачи сравнивался эффект от применения методики крупнозернистого распараллеливания с эффектом от применения методики комбинированного распараллеливания (на одном вычислительном узле). При вычислениях варьировалась только величина  $t_{max}$ , что давало в конце моделирования различное число частиц в лавине (большему значению времени соответствовало большее число частиц). Для каждого сочетания методики распараллеливания и величины  $t_{max}$  оценивались значения функции  $L = L(t)$ , где  $L(t)$  - количество реализаций лавины частиц, полученных к моменту машинного времени  $t$ .

Как показали расчеты, для  $t_{max} = 0.01$  нс в лавине образуется в среднем примерно 106 частиц, а при  $t_{max} = 1$  нс – в среднем около  $8 \cdot 10^7$  частиц. В программе, реализующей методику комбинированного распараллеливания, при числе частиц в лавине более 106 намеренно производилось периодическое перераспределение частиц между ядрами сопроцессора и «прикрепленного» CPU ядра. Как видно из рис. 3, для одной и той же величины  $t_{max}$  методика комбинированного распараллеливания позволяет получить значительно большее число реализаций в течение заданного машинного времени. Таким образом, при большой величине  $t_{max}$  методика комбинированного распараллеливания предпочтительнее методики крупнозернистого распараллеливания (несмотря на затраты машинного времени на балансировку вычислительной нагрузки).

## Заключение

В дальнейшем предполагается усовершенствовать разработанный параллельный алгоритм с целью моделирования облака электронов начиная с момента, когда собственное электрическое поле электронов и ионов лавины становится сравнимым с внешним электрическим полем [1]. С целью повышения эффективности моделирования следует, возможно, отказаться от лексикографической схемы «ветвления» траекторий и использовать «метод поколений» [2]. Следует также оценивать влияние высокоэнергичных электронов на развитие пробоя в газе [1, 2]. Эта задача связана с оценкой т.н. «редких» событий и требует применения специальных методик статистического моделирования. Для таких постановок статистическое моделирование на гибридных суперкомпьютерах с использованием методики комбинированного распараллеливания представляется весьма перспективным.

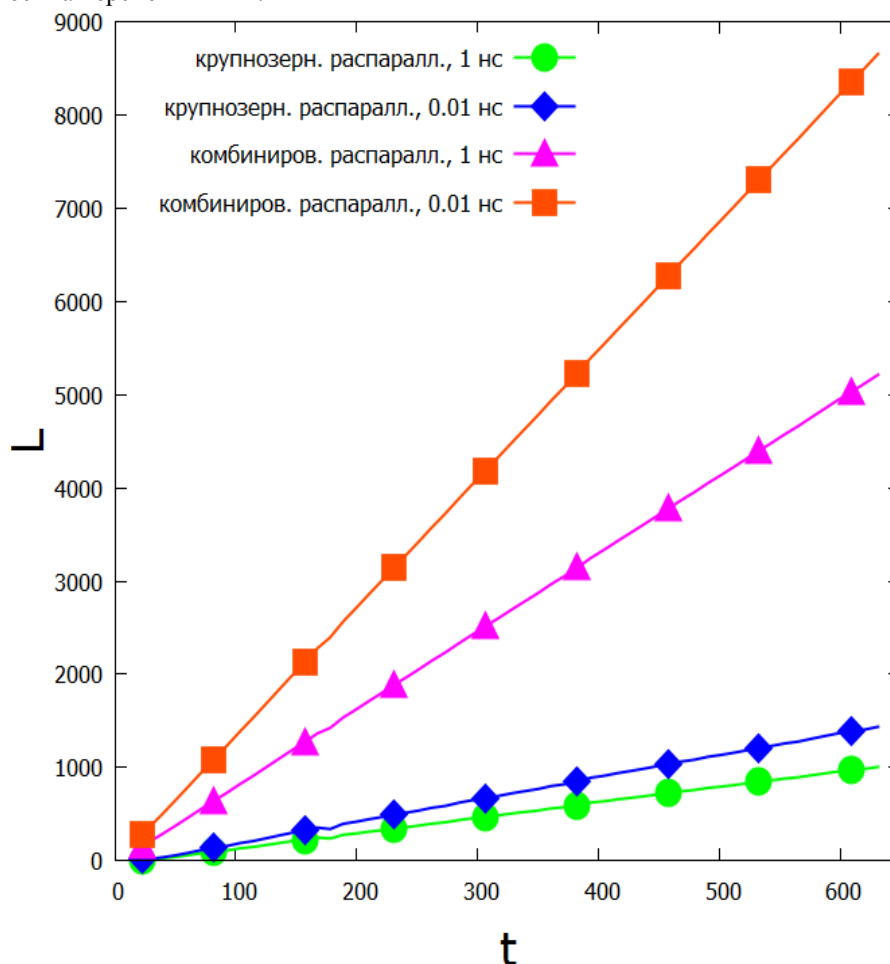


Рис. 3. Сравнение двух методик распараллеливания. По горизонтальной оси отложено машинное время  $t$  в секундах, по вертикальной оси – соответствующее количество реализаций  $L$ . Линия с ромбами - крупнозернистое распараллеливание, физическое время – 0.01 нс; линия с кругами - крупнозернистое распараллеливание, физическое время – 1 нс; линия с квадратами - комбинированное распараллеливание, физическое время – 0.01 нс; линия с треугольниками - комбинированное распараллеливание, физическое время – 1 нс

Представляется целесообразным оценивать эффективность исполнения разработанной параллельной программы на перспективных многопроцессорных вычислительных системах с большим числом ядер. С этой целью планируется исследование возможности отображения разработанной программы на различные архитектуры гибридных суперкомпьютеров методами имитационного моделирования, технология которого изложена в работе [10].

Пользуясь случаем, автор выражает признательность гл. специалисту ЦКП ССКЦ СО РАН Н.В. Кучину за помощь и плодотворные обсуждения.

Настоящая работа проводилась при финансовой поддержке грантов РФФИ №№ 13-07-00589, 13-01-00746, 12-01-00034, 12-01-00727; МИП №№ 39, 47, 126, 130 СО РАН.

ЛИТЕРАТУРА:

1. Королев Ю.Д., Месяц Г.А. Физика импульсного пробоя газов. М.: Наука, Гл. ред. физ.-мат. лит., 1991, 224 с.
2. Аккерман А.Ф. Моделирование траекторий заряженных частиц в веществе. М.: Энергоиздат, 1991.
3. Shklyaeв V. A., Belomyttsev S. Ya., Ryzhov V. V. Simulation of the formation of a runaway electron beam in an overvolted gas gap breakdown // Journal of Applied Physics, 2012, V.112, Is.11.
4. Лотова Г.З., Марченко М.А., Михайлов Г.А., Рогазинский С.В., Рыжов В.В., Ухинов С.А., Шкляев В.А. Параллельная реализация метода Монте-Карло для моделирования развития электронных лавин в газе // Известия высших учебных заведений. Физика, 2013 (в печати)
5. Жуковский М.Е., Усков Р.В. Математическое моделирование радиационной эмиссии электронов на гибридных суперкомпьютерах // Вычислительные методы и программирование, 2012, Т.13, №1, с. 189-197.
6. Марченко М.А., Михайлов Г.А. Распределенные вычисления по методу Монте-Карло // Автоматика и телемеханика. 2007, Вып. 5, С. 157–170.
7. Марченко М.А. Библиотека PARMONC для решения «больших задач по методу Монте-Карло»// Вестник ННГУ, 2012, №5, С. 392-397.
8. Марченко М.А. Страница библиотеки PARMONC на сайте ЦКП ССКЦ СО РАН: <http://www2.sscs.ru/SORAN-INTEL/paper/2011/parmonc.htm>
9. Jeffers J., Reinders J. Intel Xeon Phi Coprocessor High-Performance Programming. Elsevier, 2013, 432 pages.
10. Глинский Б.М., Родионов А.С., Марченко М.А., Караваев Д.А., Подкорытов Д.И., Винс Д.В. Исследование масштабируемости параллельных алгоритмов методом агентно-ориентированного моделирования // Параллельные вычислительные технологии (ПАВТ'2013): труды международной научной конференции (г. Челябинск, 1-5 апреля 2013 г.), 2013, С. 318-329.