

УДК 518.8:533.9

О РАСПАРАЛЛЕЛИВАНИИ МЕТОДА ЧАСТИЦ ДЛЯ ГИБРИДНОГО СУПЕРКОМПЬЮТЕРА

© 2022 г. Академик РАН Б. Н. Четверушкин¹, М. Б. Марков^{1,*}, Р. В. Усков¹

Поступило 21.03.2022 г.
После доработки 08.05.2022 г.
Принято к публикации 01.06.2022 г.

Представлена реализация алгоритма метода частиц на гибридном кластере с графическими процессорными устройствами (ГПУ) для моделирования объемной генерации, распространения и рассеяния электронов в самосогласованном электромагнитном поле. Предложены решения проблем синхронизации записи результатов и обмена данными при расчете параметров частиц, снижающие расход оперативной памяти ГПУ.

Ключевые слова: метод частиц, синхронизация вычислений, графическое процессорное устройство
DOI: 10.31857/S2686954322040063

1. ВВЕДЕНИЕ

Преимуществом гибридных суперкомпьютеров с ГПУ является высокая скорость вычислений. Она обусловлена большой пиковой производительностью ГПУ при однотипной обработке большого объема данных. Поэтому ГПУ являются перспективным средством повышения эффективности математического моделирования физических процессов [1].

Архитектура ГПУ предъявляет особые требования к вычислительным алгоритмам. Поэтому большинство научных и технических проблем, требующих численного решения уравнений математической физики, по-прежнему решаются на центральных процессорных устройствах (ЦПУ).

Данная работа представляет реализацию вычисления распределений электронов в самосогласованном электромагнитном поле методом частиц [2] на гибридном кластере. Метод подразумевает расчет электронных координат и импульсов, изменяющихся под действием поля и столкновений, совместно с решением уравнений Максвелла с плотностью тока в правой части. Гибридный кластер состоит из вычислительных узлов, в которых на каждый ГПУ приходится n штук ЦПУ. ГПУ включает m потоковых мультипроцессоров – специализированных чипов, каждый из которых является независимым вычислителем.

Узел имеет общую оперативную память, доступную всем ЦПУ. Оперативная память ГПУ доступна всем его чипам. Объем вычислений динамических переменных и плотности тока превышает потребный для решения уравнений Максвелла. Поэтому использование ГПУ для реализации метода частиц является актуальной проблемой [3–8].

Данная работа представляет реализацию метода частиц на гибридном кластере для задач со следующей спецификой. Электроны генерируются во всей расчетной области и рассеиваются с малой передачей импульса. Длина пробега, лармовский и дебаевский радиусы электронов малы по сравнению с размером расчетной области.

2. ПОСТАНОВКА ЗАДАЧИ И ЧИСЛЕННЫЙ АЛГОРИТМ

Рассмотрим кинетическое уравнение для зависящей от времени t функции распределения электронов $f = f(t, \mathbf{r}, \mathbf{p})$ в фазовом пространстве координат \mathbf{r} и импульсов \mathbf{p} [9]:

$$\frac{\partial f}{\partial t} + \operatorname{div}(\mathbf{v}f) + e \operatorname{div}_{\mathbf{p}} [(\mathbf{E} + [\mathbf{v}, \mathbf{H}]/c)f] + \sigma' v f = Q + \int d\mathbf{p}' \sigma(\mathbf{p}, \mathbf{p}') v' f(t, \mathbf{r}, \mathbf{p}'), \quad (1)$$

$$\operatorname{rot} \mathbf{H} = \frac{1}{c} \frac{\partial \mathbf{E}}{\partial t} + \frac{4\pi}{c} \mathbf{j}, \quad \operatorname{rot} \mathbf{E} = -\frac{1}{c} \frac{\partial \mathbf{H}}{\partial t}, \quad (2)$$

где \mathbf{v} – скорость, $\mathbf{E}(t, \mathbf{r})$ и $\mathbf{H}(t, \mathbf{r})$ – напряженности электрического и магнитного полей, $\operatorname{div}_{\mathbf{p}}$ – дивергенция в пространстве импульсов, e – заряд электрона, c – скорость света, σ' и $\sigma(\mathbf{p}, \mathbf{p}')$ – пол-

¹ Федеральный исследовательский центр
Институт прикладной математики им. М.В. Келдыша
Российской академии наук, Москва, Россия

*E-mail: m_b_markov@mail.ru

ное и дифференциальное макроскопические сечения рассеяния электронов, $Q(t, \mathbf{r}, \mathbf{p})$ – интенсивность генерации электронов в фазовом пространстве; $\mathbf{j}(t, \mathbf{r}) = e \int \mathbf{v} f d\mathbf{p}$ – плотность электрического тока. Интеграл столкновений в (1) моделирует тормозное излучение, упругое рассеяние электронов, ионизацию и возбуждение среды. При низких энергиях учитывается прилипание к молекулам и рекомбинации с ионами [10–12].

Основу алгоритма решения уравнения (1) составляет представление решения суммой частиц $\omega_l \delta(\mathbf{r} - \mathbf{r}_l^s) \delta(\mathbf{p} - \mathbf{p}_l^s)$, где функции $\mathbf{r} = \mathbf{r}_l^s(t)$, $\mathbf{p} = \mathbf{p}_l^s(t)$ описывают траекторию частицы [13], появившейся в момент времени t_l с начальными координатами $(\mathbf{r}_l, \mathbf{p}_l)$ в фазовом пространстве. Величина ω_l определяет количество электронов, моделируемое частицей с номером l . Параметры $t_l, \mathbf{r}_l, \mathbf{p}_l$ и ω_l выбираются так, чтобы максимально точно аппроксимировать источник электронов Q минимальным количеством частиц. Алгоритм сочетает статистическое моделирование рассеяния электронов с решением уравнений их движения в электромагнитном поле между столкновениями [14]. Пусть в моменты времени $t_l \in [t_n; t_{n+1}]$ в ячейке разностной сетки образуется L частиц, в том числе, оставшихся с предыдущего временного шага с $t_l = t_n$, генерируемых источником в течение интервала (t_n, t_{n+1}) , прибывающих из соседних ячеек и образующихся при ударной ионизации. Алгоритм вычисляет функцию $f(t_{n+1}, \mathbf{r}, \mathbf{p})$, удовлетворяющую уравнению (1) с источником

$$Q_n(t, \mathbf{r}, \mathbf{p}) = \sum_{l=1}^L \omega_l \delta(t - t_l) \delta(\mathbf{r} - \mathbf{r}_l) \delta(\mathbf{p} - \mathbf{p}_l).$$

Выполнение алгоритма состоит в построении для частицы траектории, которую она проходит за время $t_{n+1} - t_l$. При этом моделируются события: выход из ячейки, поглощение, рассеяние, генерация вторичной частицы. Траектория разбивается на сегменты между событиями. Вычисляется время t_p , за которое частица проходит сегмент. Оно определяется как минимальное из интервала $t_a = t_{n+1} - t_l$, оставшегося до конца шага по времени; времени до пересечения границы ячейки t_m ; времени до столкновения t_c . После определения t_p интегрируются уравнения движения в заданном поле. Значения его компонент интерполируются в точку, где находится частица. Рассчитывается плотность тока, созданного частицей за время t_p . Значение t_a пересчитывается: $t_a \rightarrow t_a - t_p$. Если $t_p = t_c$, то после вычисления новых координат и импульсов моделируется столкновение. Импульс частицы мгновенно изменяется статистически в соответствии с дифференциаль-

ным сечением рассеяния. Если при столкновении возникла новая частица, то она заносится в список необработанных с координатами, весом и параметром t_a рассеянной частицы. Если после окончания построения сегмента величина t_a остается положительной, то алгоритм расчета повторяется для следующего сегмента, начиная с определения t_p . Если $t_a = 0$, то частица считается достигшей верхнего временного слоя и заносится в список обработанных с текущими параметрами. Моделирование столкновений, решение уравнений движения, расчет плотности тока рассмотрены в работах [9, 14, 15].

3. РЕАЛИЗАЦИЯ АЛГОРИТМА НА ГИБРИДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ

Рассмотренный алгоритм решения уравнения (1) позволяет выделить основной объем логически простых вычислений – расчет параметров частиц, и выполнить их на ГПУ. Уравнения Максвелла (2) решаются с помощью явной разностной схемы [16] и требуют существенно меньшего объема вычислений. Поэтому задача совместного решения уравнений (1)–(2) разбивается на два этапа. Траектории частиц и плотность тока в заданном электромагнитном поле вычисляются на ГПУ под управлением ЦПУ. Затем ЦПУ вычисляют электромагнитное поле на следующем шаге. Каждое ЦПУ обрабатывает одну из прямоугольных подобластей расчетной области. Обмен данными на границах подобластей осуществляется в модели “матрица процессоров”.

Рассмотрим вычисление плотности тока. Оперативная память и чипы каждого ГПУ делятся между n ЦПУ. В части ГПУ, выделенной данному ЦПУ, обрабатываются частицы из его подобласти. На каждом временном слое ЦПУ присваивает вновь появившимся частицам параметры $\omega, \mathbf{r}_l, \mathbf{p}_l, t_l$ и передает их на ГПУ. Также на ГПУ передаются значения сеточных компонент поля на данном слое, **используемые** вместе с сечениями рассеяния для моделирования движения частицы всеми чипами. На одном временном шаге используемые данные не меняются. Кроме того, каждому чипу отводится свой участок общей памяти ГПУ для хранения **вычисляемых** данных – сеточных значений плотности тока. Это исключает синхронизацию записи вкладов частиц, обрабатываемых отдельными чипами, в вычисляемые величины. При этом объем памяти, хранящей вычисляемые данные, резко увеличивается.

Рассмотрим выполнение вычислений на ГПУ. Под вычислительным ядром понимается задание на обработку некоторой совокупности частиц. Под обработкой понимается пересчет координат и импульсов частицы при построении очередного

сегмента траектории. Ядро есть совокупность исходных данных о координатах и импульсах частиц на текущем слое, применяемого алгоритма, параллельной конфигурации вычислителей. Кодом ядра называется последовательность арифметических операций над характеристиками частицы, реализующая алгоритм для каждой из них. Вычислительные потоки, выполняющие код, группируются в блоки в соответствии с конфигурацией ядра. Конфигурация задает количество вычислителей в блоке и количество блоков. Блок потоков целиком выполняется одним чипом. Чтобы потоки могли обрабатывать независимые данные, т.е. разные частицы, при выборе данных они ориентируются на номер блока и свой номер в нем. Каждый блок в комплекте вычисляемых данных хранит значения плотности тока в ячейках сетки. В процессе обработки потоки блока атомарно записывают вклады частиц в вычисляемые данные для соответствующих ячеек. Вклады частиц блоков по окончании моделирования на данном временном слое суммируются.

За один запуск ядра для частицы строится очередной сегмент траектории. Для отдельной частицы результатом запуска может быть: переход на следующий слой и окончание моделирования на данном шаге по времени; достижение границы ячейки или точки взаимодействия; поглощение и окончание моделирования. Количество обрабатываемых при однократном запуске ядра частиц есть произведение количеств потоков в блоке $N_{\text{block}} = 512$ и числа мультипроцессоров $N_{\text{mp}} = m/n$, т.е. несколько тысяч. Для перевода всех частиц на следующий слой ядро запускается многократно. Вычислители, частицы которых после очередного запуска обработаны, при следующем запуске начинают обрабатывать другие частицы системы. Запуски ядра продолжаются до достижения всеми частицами данного ЦПУ следующего слоя.

Данные о частицах на ГПУ хранятся в трех доменах памяти. В первом домене хранятся данные частиц, оставшихся в подобласти с предыдущего шага по времени. Для каждого из потоков закладывается собственный домен под частицы, всего $N_{\text{block}} * N_{\text{mp}}$ доменов. Второй и третий домены хранят данные входящих и выходящих частиц. К входящим относятся частицы, рожденные на текущем шаге по времени, а также прилетевшие из соседних подобластей сетки. К выходящим относятся частицы, вылетевшие за границу подобласти. Для входящих и выходящих частиц создается по отдельному домену — “корзине” на каждый чип, всего N_{mp} “корзин” каждого типа. Входные корзины равномерно заполняются на стороне ЦПУ в перерывах между запусками ядра. Аналогично на стороне ЦПУ опустошаются выходные корзины. Для каждого потока на ГПУ хранятся номер I_{write} элемента памяти, из которого берутся характери-

стики частицы для обработки при запуске ядра, и номер I_{read} элемента памяти, в который будут записаны изменившиеся характеристики обработанной частицы. Для каждого чипа на ГПУ также хранятся количества частиц во выходной (I_{out}) и входной (I_{in}) корзинах.

Последовательность вычислений (рис. 1) включает фрагментацию памяти в массиве, т.е. возникновения пустых элементов. Если частица достигает конца шага по времени, оставаясь в текущей подобласти, то указатели I_{read} и I_{write} одновременно перемещаются на следующий элемент памяти. Если частица выбывает из рассмотрения и в массиве и образуется свободная ячейка, то вперед двигается только указатель I_{read} . Таким образом, все последующие частицы после обработки сдвинутся в массиве данных и закроют “дыру”. Это позволяет автоматически балансировать количество частиц между потоками. Поток забирает на обработку частицы из входных корзин только после того, как все частицы, оказавшиеся в его домене на начало шага по времени, будут обработаны. Поэтому чем больше у потока частиц для обработки, тем позднее он начнет брать дополнительные частицы из входной корзины. Наполнение корзин балансируется на ЦПУ. После окончания обработки всех частиц на всех ГПУ вклады в плотности токов от всех блоков суммируются на каждом из ГПУ. Вклады от каждой подобласти отправляются на ЦПУ для решения уравнений Максвелла на верхнем слое. Проблемой данной реализации является необходимость хранения в памяти ГПУ отдельного комплекта вычисляемых данных для каждого чипа. Уменьшить объем памяти, занимаемой комплектом, позволяет управление каждым ГПУ не одним, а несколькими (n) ЦПУ. Пусть ГПУ обрабатывает частицы в N ячейках сетки. Каждый ЦПУ управляет обработкой частиц на ячейках сетки в количестве N/n . На один ЦПУ приходится $1/n$ памяти и m/n чипов ГПУ. Тогда расход памяти ГПУ на хранение вычисляемых данных пропорционален (Nm/n^2) .

Следующей проблемой является межпроцессорная синхронизация окончания шага по времени. Она необходима, поскольку частицы могут перемещаться между подобластями в рамках одного шага. При моделировании очередного шага на каждом ЦПУ одновременно происходят два параллельных процесса: запуск на ГПУ ядер с обменом частицами и координация MPI-обменов. При координации MPI-обменов итеративно проверяется готовность ЦПУ. Считается, что оно закончило моделирование, если все частицы его подобласти достигли конца шага по времени, а для всех частиц, отправленных в соседние подобласти, получено подтверждение о получении адресатом. В случае готовности ЦПУ выставляет асинхронный барьер, не блокирующий коорди-

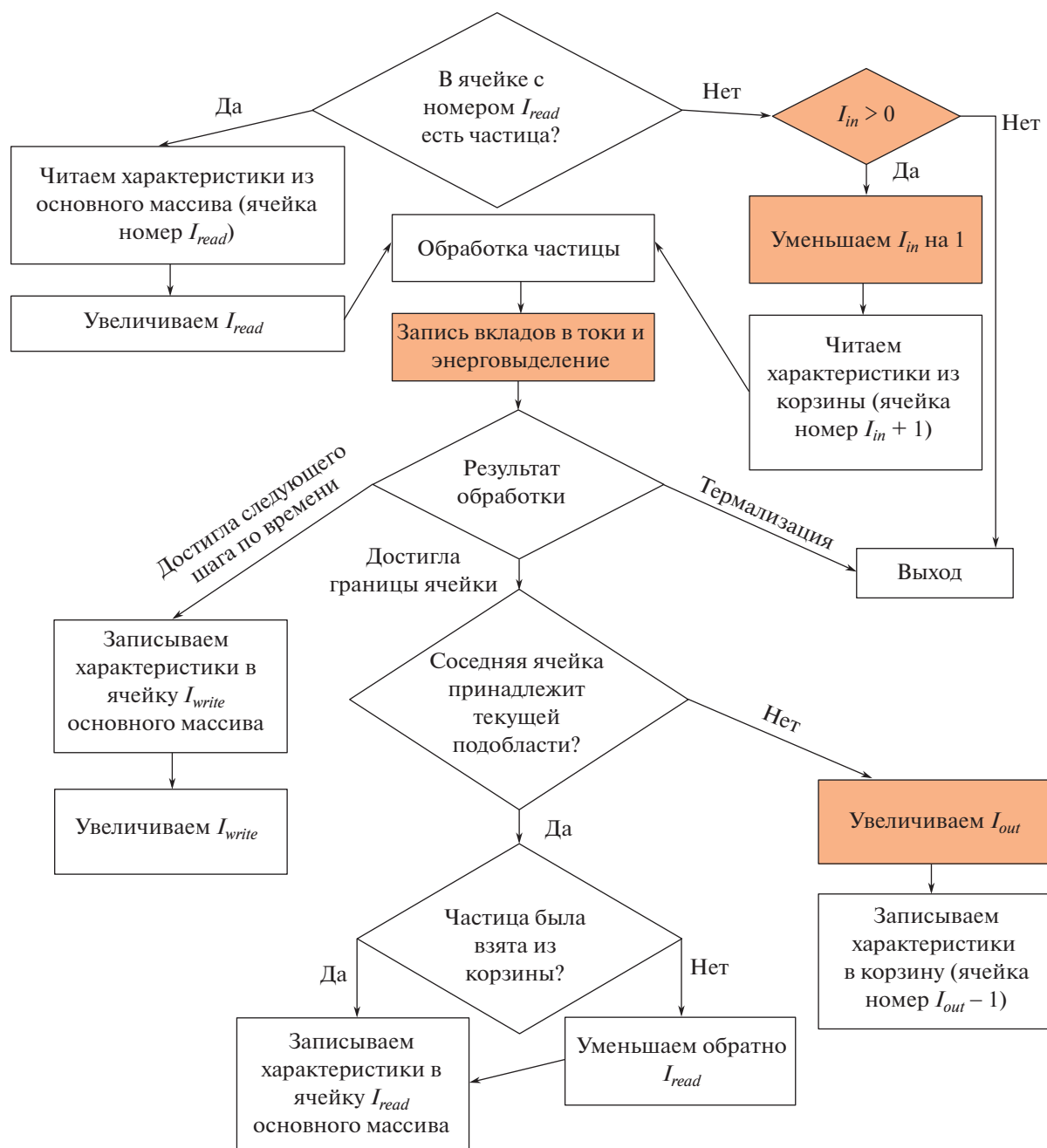


Рис. 1. Организация вычислений. Цветом отмечены операции, требующие синхронизации внутри блока потоков.

нацию MPI-обменов. Таким образом ЦПУ, закончивший моделирование, при получении новых частиц из соседней подобласти возобновляет их обработку. По достижении всеми ЦПУ барьера моделирование приостанавливается. Подсчитывается общее количество необработанных частиц. Если оно равно нулю, шаг по времени заканчивается.

Эффективность реализации решения уравнения (1) проверена в ИПМ им. М.В. Келдыша РАН на одном узле вычислительного кластера К60. Он

включает 32 ядра ЦПУ в составе двух процессоров Intel Xeon Gold 6142 v4 и четыре ГПУ nVidia V100 GV100GL, 768 Gb RAM, 2 Тб HDD. Рассмотрена задача моделирования движения в постоянном магнитном поле электронов, образуемых однородным объемным источником. Параметры подобраны так, что частицам приходится многократно пересекать границы подобластей. Сравнились скорости расчета параметров 30 млн. частиц только на ЦПУ и в рассмотренной модели распараллеливания с использованием ГПУ. По-

казано, что применение ГПУ обеспечивает ускорение более, чем в 30 раз. С уменьшением числа частиц ускорение растет. Это связано со снижением затрат ресурсов на пересылку параметров частиц между подобластями. Также уменьшение числа частиц сохраняет равномерность их распределения в пространстве, т.е. по вычислителям ГПУ.

Авторы надеются, что предложенное решение проблемы синхронизации записи результатов и обмена данными при расчете параметров частиц поможет заинтересованным специалистам при решении практических задач моделирования объемной генерации, распространения и рассеяния релятивистских электронов в самосогласованном электромагнитном поле.

СПИСОК ЛИТЕРАТУРЫ

1. Четверушкин Б.Н. Суперкомпьютерные технологии: Проблемы и перспективы ближайшего будущего // Вестн. РАН. 2018. Т. 88. № 12. С. 1083–1089.
2. Бэдсел Ч., Ленгдон А. Физика плазмы и численное моделирование: Пер. с англ. // М.: Энергоатомиздат, 1989. 452 с. Charles K. Birdsall, A. Bruce Langdon. Plasma physics, via computer simulation // McGraw-Hill Book Company, 1985.
3. Dadyka D.I., Anisimov I.O. Implementation of the Modern Plasma Simulation Codes via PIC Method for Parallel Computing Systems // Problems of Atomic Science and Technology. 2017, № 1. Series: Plasma Physics (23). P. 64–67.
4. Myers A. et al. Porting WarpX to GPU-accelerated platforms // Parallel Computing. 2021. V. 108. 102833.
5. Fatemi S. et al. AMITIS: A 3D GPU-Based Hybrid-PIC Model for Space and Plasma Physics. // IOP Conf. Series: Journal of Physics: Conf. Series. 2017. 837. 012017.
6. Романенко А.А., Снытников А.В. Оптимизация перепорядочивания модельных частиц при реализации метода частиц в ячейках на GPU // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17. № 1. С. 82–89.
7. Bastrakov S., Donchenko R., Gonoskov A., Efimenko E., Malyshev A., Meyerov I., Surmin I. Particle-in-cell plasma simulation on heterogeneous cluster systems. // Journal of Computational Science. 2013. V. 3. P. 474–479.
8. Kelling J., Bastrakov S. et al. Challenges Porting a C++ Template-Metaprogramming Abstraction Layer to Directive-based Offloading. // arXiv-CS-Software Engineering (IF), 2021-10-16.
9. Berezin A.V., Markov M.B., Parot'kin S.V. On the Particle Method for Electrons in an Inhomogeneous Scattering Medium // Computational Mathematics and Mathematical Physics. 2021. T. 61. № 9. С. 1521–1531.
10. Мэсси Г., Бархон Е. Электронные и ионные столкновения. М.: МИР, 1958. S.W. Massey, E.H.S. Burhop. Electronic and Ionic Impact Phenomena // Oxford: Clarendon Press, 1969.
11. Kim Y.-K., Rudd M.E. Theory for Ionization of Molecules by Electrons. // Phys. Rev. 1994. A 50. P. 3954–3967.
12. Александров Н.Л. Трехчастичное прилипание электрона к молекуле // УФН. 1988. V. 154. № 2. С. 177.
13. Braun W., Hepp K. The Vlasov Dynamics and Its Fluctuations in the 1/N Limit of Interacting Classical Particles. Commun. math. Phys. 1977. V. 56.
14. Berezin A.V., Vortonsov A.S., Zhukovskiy M.E., Markov M.B., Paroy'kin S.V. Partical method for electrons in a scattering medium // Comput. Math. Math. Phys. 2015. V. 55. № 9. P. 1534–1546.
15. Андрианов А.Н., Березин А.В., Воронцов А.С., Ефимкин К.Н., Марков М.Б. Моделирование электромагнитных полей радиационного происхождения на многопроцессорных вычислительных системах. // Препринт ИПМ им. М.В. Келдыша РАН № 74, 2006.
16. Березин А.В., Марков М.Б., Плющенко Б.Д. Локально-одномерная разностная схема для электродинамических задач с заданным волновым фронтом. // Препринт ИПМ им. М.В. Келдыша РАН № 31, 2005.

ON THE PARALLELIZATION OF THE PARTICLE METHOD FOR HYBRID SUPERCOMPUTER

Academician of the RAS B. N. Chetverushkin^a, M. B. Markov^a, and R. V. Uskov^a

^a Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Moscow, Russian Federation

An implementation of the particle method algorithm on a hybrid cluster with graphic processor units (GPU) for modeling volumetric generation, propagation and scattering of electrons in a self-consistent electromagnetic field is presented. Solutions to the problems of synchronization of recording results and data exchange in the calculation of particle parameters, which reduce the consumption of GPU RAM, are considered.

Keywords: particle method, synchronization of calculations, GPU