Трехмерное моделирование плазмы методом частиц в ячейках на Intel Xeon Phi: оптимизация вычислений и примеры использования[1]

И.Б. Мееров[1], С.И. Бастраков[1], И.А. Сурмин[1], А.А. Гоносков[1,2], Е.С. Ефименко[1,2], А.В. Башинов[1,2], А.В. Коржиманов[1,2], А.В. Ларин[1], А.А. Муравьев[1,2], А.И. Розанов[1], М.Р. Савичев[1]

[1] Нижегородский государственный университет им. Н.И. Лобачевского,
[2] Институт прикладной физики РАН

Рассматривается проблема эффективного использования ускорителей Xeon Phi при моделировании лазерной плазмы. Приводится анализ особенностей архитектуры Xeon Phi, влияющих на производительность кода при численном моделировании плазмы методом частиц в ячейках. Описывается параллельный программный комплекс PICADOR, оптимизированный ранее для расчетов на ускорителях. Производительность программного комплекса на Xeon Phi в сравнении с CPU исследуется при решении трех вычислительно трудоемких задач. Дается соотношение времени расчета на Xeon Phi и CPU на разных этапах метода частиц в ячейках. Демонстрируется, что в зависимости от особенностей задачи Xeon Phi может как опережать, так и отставать от CPU при выполнении расчетов.

1. Введение

Численное моделирование лазерной плазмы – одна из актуальных областей современной вычислительной физики, имеющая обширное практическое применение. Среди важных приложений можно отметить проектирование компактных источников для адронной терапии при лечении онкологических заболеваний, создание фабрик короткоживущих изотопов для биоимиджинга, разработку приборов для исследования внутримолекулярных и внутриатомных процессов. Для численного моделирования в данной области широко применяется метод частиц в ячейках [1] (метод макрочастиц, Particle-In-Cell, PIC). Одной из основных целей ученых, развивающих данный метод в контексте моделирования плазмы [2, 3], является повышение точности расчетов до уровня, достаточного для замены как можно большого числа натурных экспериментов вычислительными.

Суперкомпьютерные системы играют значительную роль в достижении указанной цели. Так, метод PIC представляет плазму в виде совокупности большого количества взаимодействующих макрочастиц, объединяющих частицы с близкими свойствами. В настоящее время известны актуальные для науки задачи, для достижения приемлемой точности в которых требуется использование $\sim 10^{10}$ макрочастиц, расположенных в $\sim 10^9$ ячейках пространственной сетки, что определяет актуальность применения суперкомпьютеров и разработки соответствующего программного обеспечения (ПО). Работы по созданию такого ПО ведутся научными группами в России [4–10] и за рубежом [11–14]. Существуют и активно применяются программные комплексы (PIC-коды), реализующие метод частиц в ячейках и ориентированные на традиционные (OSIRIS [11], VLPL [12], VPIC [13]) и гетерогенные (PIConGPU [14]) кластерные системы. С 2010 года авторами разрабатывается программный комплекс PICADOR [15, 16], имеющий реализации широкого спектра численных схем, оптимизированных для современных кластерных систем с использованием методов динамической балансировки нагрузки [17]. PICADOR допускает использование ускорителей GPU и Xeon Phi [18]. Функциональность может быть расширена посредством специализированного интерфейса Module Development Kit (MDK) [19]. Проект развивается в сотрудничестве с исследовательскими группами из ИПФ РАН (Нижний Новгород) и Чалмерского технологического университета (Гетеборг, Швеция).

[1] Работа частично поддержана РФФИ, грант 14-07-31211, и грантом МОН РФ (соглашение от 27 августа 2013 г. № 02.B.49.21.0003 между МОН РФ и ННГУ им. Н.И. Лобачевского).

Данная работа посвящена вопросам эффективной реализации метода частиц в ячейках для ускорителей Intel Xeon Phi, выполненной в контексте программного комплекса PICADOR. Первые шаги в данном направлении были сделаны нами в работах [18, 20]. Было показано, что применение различных подходов к оптимизации расчетов (улучшение локальности обращений в память, частичная векторизация и др.) при решении модельной задачи при *50 макрочастицах на ячейку* позволяет достичь *7,7 наносекунд на макрочастицу* на ускорителе Intel Xeon Phi 5110P (60 ядер, 240 потоков, 8 ГБ ОЗУ) при базовом значении *14 наносекунд на макрочастицу* на Intel Xeon E5-2660 с частотой 2,2 ГГц (Sandy Bridge, 8 ядер, 20 МБ кэш-памяти, поддержка AVX). Кроме того, необходимо отметить работу [21], основным результатом которой также является оптимизация для Xeon Phi одной из реализаций метода частиц в ячейках. При решении модельной задачи, содержащей вдвое большее число макрочастиц на ячейку, на ускорителе Intel Xeon Phi достигнуто *1,7 наносекунд на макрочастицу*. Данные результаты нельзя сопоставлять напрямую по следующим причинам: в отличие от широко применяемой в PIC-моделировании сетки Йи (Yee), в работе [21] использована прямая сетка, кардинально упрощающая векторизацию вычислений, а также использовано вдвое большее число частиц, что повышает эффективность использования ресурсов.

В данной работе мы делаем следующий шаг в изучении производительности комплекса PICADOR на Xeon Phi в реальных условиях. Для этого мы рассматриваем следующие задачи, представляющие научный интерес: лазерное ускорение протонов в тонких мишенях с субволновыми неоднородностями на облучаемой стороне, компрессия лазерных импульсов на кильватерной плазменной волне, генерация гамма-излучения в сходящейся дипольной волне. Значения параметров расчета, влияющих на производительность кода, выбираются исходя из физики задачи. Основное внимание уделяется сравнению производительности кода на CPU и Xeon Phi и выделению особенностей задачи, влияющих на время расчета.

2. Краткая характеристика вычислительной схемы метода частиц в ячейках

В данном разделе приводится краткое описание вычислительной схемы метода частиц в ячейках в соответствии с [18], подробное описание метода содержится в [1].

Область моделирования имеет форму прямоугольного параллелепипеда со сторонами, параллельными осям декартовой системы координат. В расчетной области заданы электрическое и магнитное поля E и B, динамика электромагнитного поля описывается системой уравнений Максвелла. В методе частиц в ячейках плазма моделируется набором из N заряженных макрочастиц, каждая из которых характеризуется переменными импульсом p и координатами r, а также постоянными массой m и зарядом q. Координаты и скорость частиц v меняются согласно релятивистским уравнениям движения. Движение заряженных частиц создает плазменные токи j, которые входят в качестве источников в уравнения Максвелла, таким образом, замыкая самосогласованную систему уравнений.

Рис. 1. Вычислительная схема метода частиц в ячейках. Итерация цикла соответствует моделированию одного шага по времени. Уравнения приведены в системе СГСЭ. Надписи над стрелками показывают зависимости по данным между этапами метода [18]

Вычислительная схема метода с основными уравнениями и схемой зависимости по данным приведена на рисунке 1. Начальные условия состоят из значений электрического и магнитного полей, а также координат и скоростей частиц в начальный момент времени. На каждой итерации по времени по текущим данным частиц и полей вычисляется состояние системы в следующий момент времени. Каждая итерация состоит из 4 основных этапов: интегрирование уравнений Максвелла, интерполяция полей в точке нахождения каждой частицы и вычисление силы Лоренца, действующей на частицу, интегрирование уравнений движения частиц, вычисление («взвешивание») токов. При программной реализации вычисление силы Лоренца и интегрирование уравнений движения обычно объединяются в целях повышения производительности.

3. Факторы, влияющие на производительность PIC-кодов

Главным фактором, влияющим на производительность и масштабируемость PIC-кодов и используемых в них реализаций численных схем, является эффективность использования ресурсов параллелизма на разных уровнях. Требуют решения вопросы организации параллельной обработки *на уровне распределенной памяти* (узлы кластера), *на уровне общей памяти* (вычислительные ядра в рамках одного узла), *на уровне SIMD-инструкций* (в рамках каждого ядра процессора), отдельного обсуждения заслуживает вопрос об *использовании ускорителей*.

Отметим, что потенциальная возможность параллельных расчетов заложена в основу метода частиц в ячейках. Действительно, в методе PIC макрочастицы не взаимодействуют друг с другом напрямую, и операции частица-сетка являются пространственно локальными. Используя данные факторы, разработчики ПО разделяют расчетную область на домены по территориальному принципу и организуют обмены между соседними доменами. Сочетание этого подхода с эффективными схемами балансировки нагрузки позволяет при решении ряда задач достигать приемлемой масштабируемости при использовании тысяч узлов кластера [11, 22].

При работе в рамках одного узла перед программистами вновь встает задача балансировки, однако, как показывает практика, она решается достаточно успешно базовыми средствами OpenMP, что в сочетании с использованием реализаций SMT (simultaneous multithreading) в современных процессорах позволяет добиться близкой к единице эффективности масштабируемости относительно однопоточной реализации [18].

К сожалению, при переходе к следующему уровню параллелизма (SIMD) возникают значительные трудности [18, 20, 21]. Причины этих трудностей заключены в самой природе метода, одновременно обрабатывающего большие массивы данных, содержащие информацию о координатах и скоростях макрочастиц, а также об электромагнитном поле, заданном в узлах сетки, дискретизирующей трехмерную область моделирования. В силу особенностей вычислительной схемы метода для разных ее этапов целесообразно применять разные схемы расположения в

памяти данных о полях и макрочастицах. Это соображение справедливо как с точки зрения эффективности работы с памятью (минимизация количества кэш-промахов), так и в контексте сокращения накладных расходов на упаковку/распаковку данных при работе с векторными командами. Насколько известно авторам, на сегодняшний день нет универсального решения данной проблемы, хотя в ряде частных случаев удается добиться хороших результатов [23].

Достижение приемлемой производительности вычислений на ускорителях GPU также вызывает большой интерес. Так, при реализации на GPU возникает проблема эффективной организации доступа к памяти в условиях работы тысяч потоков и малого объема кэш-памяти. Некоторые методы решения указанных проблем описаны авторами PIC-кодов [14, 22, 23].

Остановимся подробнее на факторах, ограничивающих производительность PIC-кодов на Xeon Phi. Сопроцессор Intel Xeon Phi – ускоритель вычислений общего назначения, реализован в виде отдельной платы, присоединяемой к базовой вычислительной системе через слот PCI Express. Данный недостаток Xeon Phi в сочетании с ограниченным объемом оперативной памяти на сопроцессоре (8–16 ГБ в зависимости от модели) может привести к низкой производительности из-за необходимости копирования данных между основной системой и ускорителем. Применительно к PIC-кодам эта проблема решается за счет использования достаточно большого числа узлов кластера, что, по крайней мере в решаемых нами задачах, позволяет уложиться в заявленные объемы памяти. Xeon Phi содержит 61 вычислительное ядро с аппаратной поддержкой до 244 потоков команд. Каждое ядро является полнофункциональным, но облегченным (не поддерживает внеочередное выполнение, прогноз ветвлений и другие усовершенствования, значительно влияющие на производительность). В то же время, ядра Xeon Phi поддерживают специально разработанный векторный набор команд, для обработки которых предназначен 512-битный блок векторных вычислений. Блок содержит 32 специальных 512-битных регистра и допускает выполнение за 1 такт до 8 операций над числами плавающей точкой двойной точности (в два раза больше, чем в AVX). Одной из отличительных особенностей системы команд является аппаратная поддержка операций fused multiply-add (FMA), реализующих операцию $x = x + a \cdot b$ без промежуточного округления. Пиковая производительность при расчетах в двойной точности составляет около 1 TFLOPS, что приблизительно в 5 раз превосходит пиковую производительность современных CPU.

Рассмотрим, какой вклад в производительность PIC-кодов вносят отмеченные выше особенности архитектуры. Так, уменьшение производительности ядер составляет потенциальную проблему в связи с тем, что, например, наличие FMA-инструкций, дающих вклад в размере 50% от пиковой производительности, далеко не всегда может быть эффективно использовано в реализации численных схем. В то же время, практика показывает, что наличие большого количества ядер с аппаратной поддержкой SMT можно задействовать достаточно эффективно, по крайней мере, в тех задачах, где число частиц на ячейку является достаточно большим (далее будет показано, что при решении некоторых задач оптимизированным кодом Xeon Phi начинает опережать Xeon по скорости расчетов, начиная по крайней мере с 10 частиц на ячейку).

С нашей точки зрения, основным фактором, лимитирующим эффективность использования PIC-кодов на Xeon Phi, является сложность векторизации кода. Данная проблема, описанная выше в контексте программирования для CPU, еще в большей степени проявляется на Xeon Phi, обладающем более вместительными векторными регистрами. Код, который плохо векторизуется при длине вектора 4, обычно еще хуже векторизуется при длине вектора 8. Тем не менее, применение специальных схем векторизации кода с использованием интринсиков в сочетании с достаточно хорошей эффективностью масштабируемости на общей памяти при большом числе потоков позволяет частично решить проблему, получив в итоге выигрыш по времени по сравнению с CPU. Учитывая схожесть подходов к оптимизации вычислений для CPU и Xeon Phi, сравнительно малые трудозатраты на адаптацию кода для Xeon Phi, а также перспективы появления более производительной редакции ускорителя делают его достаточно перспективным устройством для численного моделирования плазмы.

4. Краткое описание программного комплекса PICADOR

Программный комплекс для моделирования плазмы PICADOR ориентирован на решение больших задач в области моделирования лазерной плазмы на гетерогенных кластерных систе-

мах с использованием многоядерных центральных процессоров, графических процессоров и сопроцессоров Intel Xeon Phi. PICADOR использует метод частиц в ячейках и его расширения для повышения точности и учета дополнительных физических эффектов. Поддерживаются конечно-разностные схемы FDTD (finite difference time domain) [24] и NDF (numerical dispersion free) [12] для численного интегрирования уравнений поля, генерация лазерного импульса на границе, периодические и поглощающие граничные условия [25], метод Бориса для интегрирования уравнений движения [1], формфакторы частиц первого и второго порядка, схемы взвешивания токов Есиркепова [26] и Вилласенора-Бунемана [27], бегущее окно, а также динамическая балансировка нагрузки [17].

Программная реализация выполнена на языке C++ с использованием объектно-ориентированной методологии. При проектировании программного комплекса предусмотрена возможность расширения кода новыми модулями, реализующими дополнительные численные схемы (интерфейс MDK, Module Development Kit [19]). В 2014–2015 гг. с использованием указанного интерфейса реализованы и успешно интегрированы в PICADOR несколько численных схем. Для достижения лучшей производительности ядро кода, содержащее наиболее вычислительно трудоемкие операции, реализовано с использованием низкоуровневых структур данных.

Расчетная часть кода является полностью параллельной. Реализация для кластерных систем выполнена с использованием технологии MPI. Расчеты внутри одного узла распараллелены при помощи технологии OpenMP. Векторизация кода для использования SIMD-расширений центральных процессоров достигается при помощи оптимизирующих компиляторов Intel C/C++ Compiler либо GCC. Программная реализация для графических процессоров выполнена при помощи NVIDIA CUDA. Реализация оптимизирована с использованием возможностей различных уровней памяти GPU и высокоэффективных атомарных операциях в современных графических процессорах. Перенос вычислений на Intel Xeon Phi в основном не требовал значительной модификации кода. Так, многие оптимизации кода, изначально выполняемые для повышения производительности на Xeon Phi, сокращали время работы программы и на центральном процессоре. Тем не менее, для лучшей векторизации численных схем на Xeon Phi были разработаны низкоуровневые реализации с использованием интринсиков.

При выполнении расчетов PICADOR показывает следующие результаты производительности и масштабируемости. Эффективность сильной масштабируемости на распределенной памяти PICADOR достигает 70% при переходе от 16 до 2048 ядер CPU. Эффективность сильной масштабируемости на общей памяти в некоторых задачах достигает 99% при переходе от 1 до 16 ядер CPU, для Xeon Phi – 78% при переходе от 1 до 60 ядер. Ускорение при использовании сопроцессора Xeon Phi составляет около 2 раз по сравнению с 8-ядерным центральным процессором. Все перечисленные результаты получены при выполнении расчетов в двойной точности.

Программный комплекс PICADOR активно используется в научных расчетах. С его помощью был решён ряд актуальных задач современной физики плазмы. В частности, была показана возможность эффективной генерации пучка высокоэнергичных многозарядных ионов при облучении структурированных твердотельных мишеней, был обнаружен новый режим взаимодействия в условиях доминирования радиационных потерь – так называемый аномальный радиационный захват, – был проведён ряд работ по исследованию динамики взаимодействия в условиях развития квантовоэлектродинамических каскадов – электрон-позитронных лавин, рождающихся в сверхсильных полях. Расчеты ведутся с использованием суперкомпьютеров МВС-100К, МВС-10П (МСЦ РАН), «Ломоносов» (МГУ), «Лобачевский» (ННГУ).

5. Тестовая инфраструктура и способ оценки производительности

При выполнении экспериментов использовался узел кластера ННГУ «Лобачевский», содержащий 2 процессора Intel Sandy Bridge E5-2660 2.2 ГГц (8 ядер на каждом CPU), 64 ГБ оперативной памяти и 2 сопроцессора Intel Xeon Phi 5110P (60 ядер, 240 потоков, память 8 ГБ). Задачи запускались либо на двух CPU с использованием всех доступных вычислительных ядер (2 процесса по 8 потоков в каждом), либо на двух Xeon Phi (2 процесса по 240 потоков в каждом). Предварительно было установлено, что данный режим запуска в большинстве случаев является оптимальным для текущей реализации. Сравнение производительности выполнялось на каждом этапе вычислительной схемы и в целом, отдельно замерялось время обменов данными. При

обсуждении задач и достигнутых результатов обращалось внимание на следующие *факторы*, влияющие на производительность: *число макрочастиц на ячейку*; наличие или отсутствие *дисбаланса нагрузки между процессами* в результате разбиения расчетной области на домены; наличие или отсутствие *дисбаланса между потоками* в одном домене.

На графиках производительности используются следующие краткие обозначения этапов и расширений метода частиц в ячейках: «Токи» – взвешивание токов, «Частицы» – интерполяция поля и движение частиц, «FDTD» – интегрирование уравнений поля, «PML» – поглощающие граничные условия, «Генератор» – генерация импульса на границе расчетной области, «QED» – рождение фотонов и электрон-позитронных пар, «Обмены» – общее время обменов данными между узлами, «Вычисление» – общее время вычислительной части.

6. Лазерное ускорение протонов в тонких мишенях с субволновыми неоднородностями на облучаемой стороне. Моделирование на Xeon Phi

Одной из актуальных проблем современной физики взаимодействия лазерного излучения сверхвысокой интенсивности с веществом является генерация и ускорение пучков протонов и других ионов [28]. При относительно небольших интенсивностях излучения, не превышающих 10^{20} Вт/см^2, наиболее эффективным способом ускорения является так называемый метод TNSA (target normal sheath acceleration) [29]. В этом методе ускорение происходит с обратной стороны тонкой (толщиной от нескольких нанометров до нескольких микрон) металлической фольги или пластиковой пленки, облучаемой сфокусированным лазерным импульсом. Ускорение происходит за счет квазистатического поля, возникающего при разлете облака электронов, нагретых лазерным излучением до релятивистских температур.

Несмотря на простоту реализации, метод TNSA, однако, обладает относительно невысокой эффективностью – даже в наилучших условиях в энергию ускоряемых протонов идёт не более нескольких процентов энергии падающего лазерного импульса [30]. В последнее время обсуждается несколько способов увеличения эффективности. Один из них заключается в том, чтобы нанести на облучаемую поверхность мишени субволновые неоднородности, которые улучшили бы поглощение лазерного излучения и тем самым увеличили бы количество энергии, передаваемой частицам [31]. Эта задача анализировалась в ряде работ путем двумерного моделирования. Однако для анализа ситуации в реальном эксперименте, особенно в случае неоднородностей сложной формы, необходимо проведение полномасштабного трехмерного моделирования, которое требует большого количества вычислительных ресурсов. По этой причине такое моделирование может в значительной степени выиграть от использования Xeon Phi. Для демонстрационных целей предполагалось, что лазерный импульс падает нормально на мишень и имеет относительно широкий поперечный размер. Это позволило решать задачу в предположении представления импульса в виде плоской волны. Облучаемая мишень представляла собой фольгу толщиной 0,3 мкм, состоящую из ионов золота Au_{197}^{+31}, компенсированных соответствующим количеством электронов. Концентрация электронов в слое составляла $3\cdot10^{21}$ см$^{-3}$. Слой ускоряемых протонов располагался на обратной стороне мишени и имел толщину 0,1 мкм. На облучаемую поверхность мишени были нанесены периодические прямоугольные неоднородности, образующие так называемую наногребенку. Период наногребенки составлял 0,5 мкм, высота – 0,3 мкм, а толщина отдельного выступа – 0,15 мкм. Начальная температура плазмы равнялась 100 эВ, что является обычным значением для столкновительного нагрева в поле мощного лазерного излучения.

Как было отмечено выше, лазерный импульс предполагался бесконечным в поперечном направлении за счет использования периодических граничных условий. В продольном направлении он имел форму кривой Гаусса с шириной на полувысоте по амплитуде равной 42 фс. Длина волны излучения равнялась 1 мкм. Интенсивность излучения в максимуме достигала значения $3{,}75\cdot10^{19}$ Вт/см^2. Такие параметры являются типичными для титан-сапфировых лазерных систем пиковой мощностью в десятки тераватт, широко распространенных в современных лабораториях. Расчетная область имела физические размеры 12×1×1 мкм, а соответствующая сетка – 512×64×64 ячеек. Шаг по времени составил 0,026 фс, а полное время расчёта – 300 фс, таким образом, было совершено 11 512 шагов по времени.

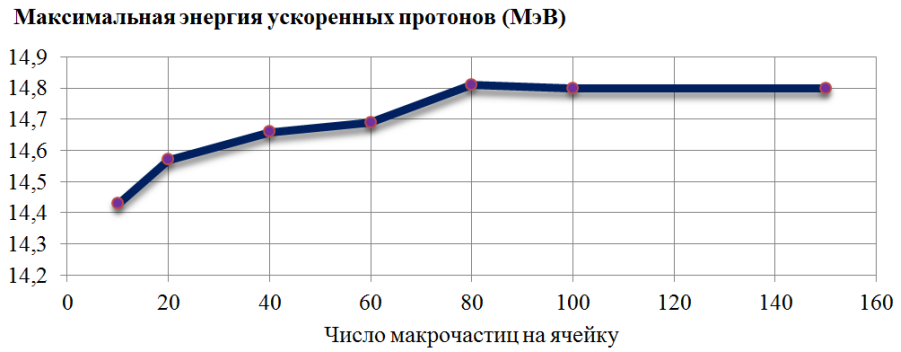

Рис. 2. Зависимость максимальной энергии ускоренных протонов от числа макрочастиц на ячейку

Одним из важнейших параметров PIC-моделирования является число макрочастиц на ячейку (*particles per cell, ppc*). Обсуждаемая задача отличается существенным влиянием, которое оказывает данный параметр на результаты моделирования. Как показано на рисунке 2, при варьировании *ppc* от 10 до 80 наблюдается непрерывный рост максимальной энергии ускоренных протонов в конце расчета, пока он не достигает своего максимума в 14,81 МэВ. Это объясняется тем фактом, что малого числа частиц не хватает для разрешения «хвоста» функции распределения нагретых электронов по энергиям. В то же время известно, что именно «хвост» этого распределения определяет максимальную энергию, которую может получить ускоренный протон. Учитывая тот факт, что начиная с 80 макрочастиц на ячейку рост максимальной энергии прекращается, именно это значение лучше всего использовать для численного моделирования.

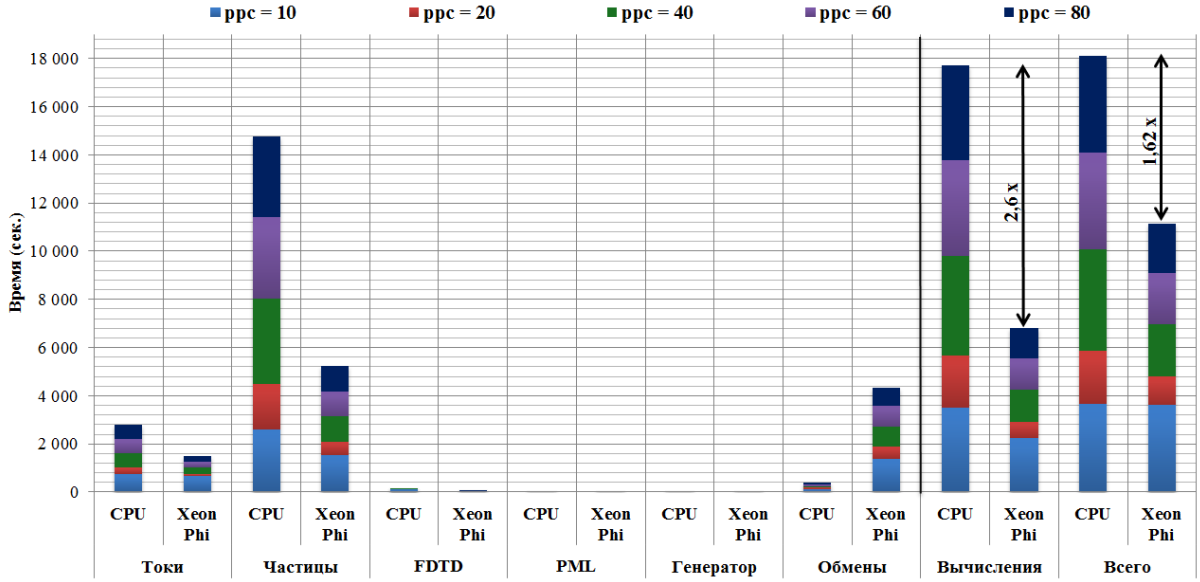

Рис. 3. Время расчета на разных этапах метода частиц в ячейках с использованием двух CPU или двух Xeon Phi. Число макрочастиц на ячейку ***ppc*** варьируется от 10 до 80. Столбцы, соответствующие времени расчета при разном значении ***ppc***, наложены друг на друга

Проанализируем производительность приложения при изменении числа макрочастиц на ячейку (*ppc*) от 10 до 80 (рисунок 3). Из диаграммы видно, что независимо от значения ppc Xeon Phi уверенно опережает Xeon как по времени вычислительной части (до 2,6 раза при *ppc* = 80), так и по общему времени счета (до 1,62 раза при *ppc* = 80), причем с ростом *ppc* преимущество ожидаемо возрастает вместе с объемом вычислений. Данная картина наблюдается на всех этапах расчета за исключением обменов, где Xeon Phi в несколько раз отстает от CPU. Данное соотношение времен в основном обусловлено необходимостью обмена информацией между MPI-процессами. При работе в рамках одного узла обмен данными между CPU реализуется средствами MPI на общей памяти, тогда как при использовании Xeon Phi требуются до-

полнительные обмены данными между ускорителем и CPU. В целом при решении данной задачи реализация для Xeon Phi показывает свою эффективность.

7. Компрессия лазерных импульсов на кильватерной плазменной волне

Другой задачей, на которой проводилось сравнение производительности CPU и Xeon Phi, является компрессия фемтосекундных лазерных импульсов до предельно-коротких длительностей (1–2 периода поля). В настоящее время хорошо освоена технология генерации фемтосекундных лазерных импульсов с длиной волны 0,8–1 мкм петаваттного уровня мощности [32]. Длительность импульсов на выходе из лазерной системы составляет около 10 периодов поля. Однако в ряде приложений, таких как генерация аттосекундных импульсов и ускорение частиц, требуются мощные предельно короткие лазерные импульсы. Генерация таких импульсов в рамках лазерной системы является труднопреодолимой задачей, поэтому необходимы другие способы укорочения лазерных импульсов. Одним из способов является компрессия фемтосекундных импульсов кильватерной плазменной волной, возбуждаемой самим импульсом [33]. Для создания плазмы используется струя газа, который ионизируется под действием лазерного импульса. Ввиду неоднородности распределения электронов в плазменной волне, передний фронт импульса движется в более плотной плазме, имеет меньшую скорость, чем задний фронт, двигающийся в менее плотной плазме. Такое взаимодействие приводит к компрессии и формированию резкого переднего фронта импульса. Важное и недостаточно изученное влияние накладывает эффект нестационарной самофокусировки коротких импульсов [34], ввиду чего необходимо использование трехмерного моделирования. При численном моделировании данной задачи необходимо использовать большое число макрочастиц, чтобы исключить влияние шумов пространственного распределения плазмы, определяющего компрессию.

Для моделирования описанного взаимодействия использовался линейно-поляризованный импульс с огибающей $\sin^2$, длительностью 30 фс по полуширине интенсивности и супергауссовым поперечным профилем с диаметром 10 мкм. Длина волны составляла 0,8 мкм, пиковая интенсивность являлась равной $3{,}3\cdot10^{19}$ Вт/см^2. Лазерный импульс фокусируется на однородную плазму с концентрацией $8\cdot10^{18}$ см$^{-3}$. Расчетная область 20×24×24 мкм разбивалась на 256×128×128 ячеек. Количество макрочастиц на ячейку равнялось 20.

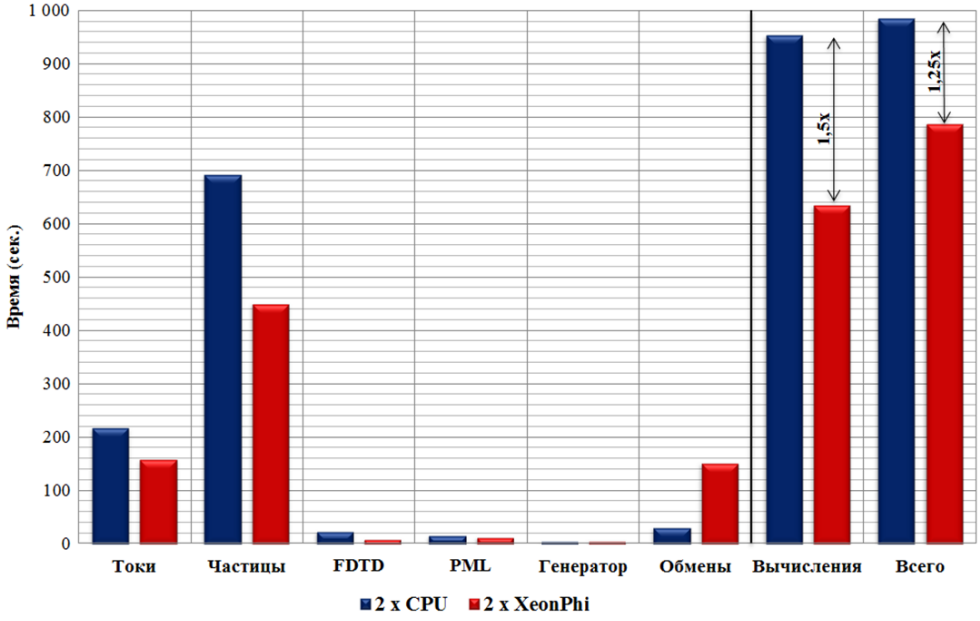

Рис. 4. Время расчета на разных этапах метода частиц в ячейках с использованием двух CPU или двух Xeon Phi. Число макрочастиц на ячейку ***ppc*** = 20

Полученные времена счета на двух Xeon Phi по сравнению с двумя CPU представлены на рисунке 4. Как и в предыдущей задаче, Xeon Phi опережает CPU на всех наиболее трудоемких

этапах расчета, кроме обменов. Выигрыш по времени работы вычислительной части составляет 1,5 раза, тогда как общее время расчета с учетом обменов данными лучше для Xeon Phi в 1,25 раза. На первый взгляд может сложиться впечатление, что данные показатели выглядят существенно скромнее, чем итоговые результаты в предыдущей задаче, но это в полной мере объясняется в 4 раза меньшим числом частиц, задействованных в расчетах. Действительно, в предыдущей задаче при *ppc* = 20 два Xeon Phi обгоняют два CPU в 1,21 раза, что весьма близко к обсуждаемому результату. В целом необходимо отметить, что использование реализации для Xeon Phi для решения данной задачи является целесообразным.

8. Генерация гамма-излучения в сходящейся дипольной волне

В сверхсильных электромагнитных полях (>10^{23} Вт/см^2) становятся возможными принципиально новые процессы (испускание жестких фотонов и рождение такими фотонами электрон-позитронных пар), которые открывают новые режимы взаимодействия в плазме, в частности каскадную генерацию частиц [35]. Подобные режимы планируется достичь в рамках проекта XCELS [36] при взаимодействии релятивистски сильных лазерных полей с плазменными мишенями. Отметим, что гамма-излучение находит большое применение в ядерной физике: получение радиоизотопов для медицины, неинвазивный анализ ядерных отходов, генерация нейтронных пучков.

Дипольная волна представляет собой инвертированное излучение диполя и реализует идеальную фокусировку лазерного излучения. Данную структуру поля в рамках проекта XCELS планируется моделировать в виде 12 сходящихся лазерных пучков, позволяющих достичь в фокусе интенсивностей на уровне 10^{26} Вт/см^2. В рассматриваемой задаче дипольная волна в форме полубесконечного импульса с резким передним фронтом падает на плазменную мишень. На начальном этапе мишень сжимается в центр расчетной области. После этого при превышении порогового значения интенсивности лазерного поля развивается электромагнитный каскад, в процессе которого число электронов, позитронов и фотонов в области каскада растет экспоненциально. Такое взаимодействие перспективно с точки зрения эффективной конверсии лазерной энергии в энергию узконаправленного гамма-излучения [37].

Для моделирования описанного взаимодействия использовалась полубесконечная дипольная волна с резким передним фронтом. Длина волны составляла $\lambda = 0{,}9$ мкм, мощность дипольной волны – 40 ПВт. Лазерный импульс фокусируется на однородную плазменную мишень диаметром 3 длины волны с концентрацией 10^{16} см$^{-3}$. Расчетная область 4×4×4 мкм разбивается на 256×256×256 ячеек. Количество макрочастиц на ячейку в начале расчета равно 20, однако в процессе развития каскада данный параметр сильно растет. При этом каскад идет в сильно ограниченной области размером порядка $0.2\ \lambda \times 0.2\ \lambda \times 0.5\ \lambda$ вблизи центра расчетной области, что приводит *к сильно неоднородному распределению частиц по ячейкам*.

Отметим, что неконтролируемое увеличение числа частиц может привести к неудовлетворительным затратам памяти и чрезмерному времени расчета. Для нивелирования данного эффекта при превышении некоторого порогового числа частиц в домене запускается процедура объединения частиц с близкими свойствами. Процессы генерации новых и объединения существующих частиц выполняются в рамках одного из этапов расчетного цикла (QED), не задействованного при решении первых двух задач.

На рисунке 5 представлено сравнение времени работы двух Xeon Phi и двух CPU при решении данной задачи. Было обнаружено, что появившийся дополнительный этап расчета занимает существенное время как на CPU, так и на Xeon Phi. В целом результаты показывают, что данная задача не является перспективной с точки зрения использования текущей реализации для Xeon Phi в связи с ее отставанием по времени в 1,4 раза.

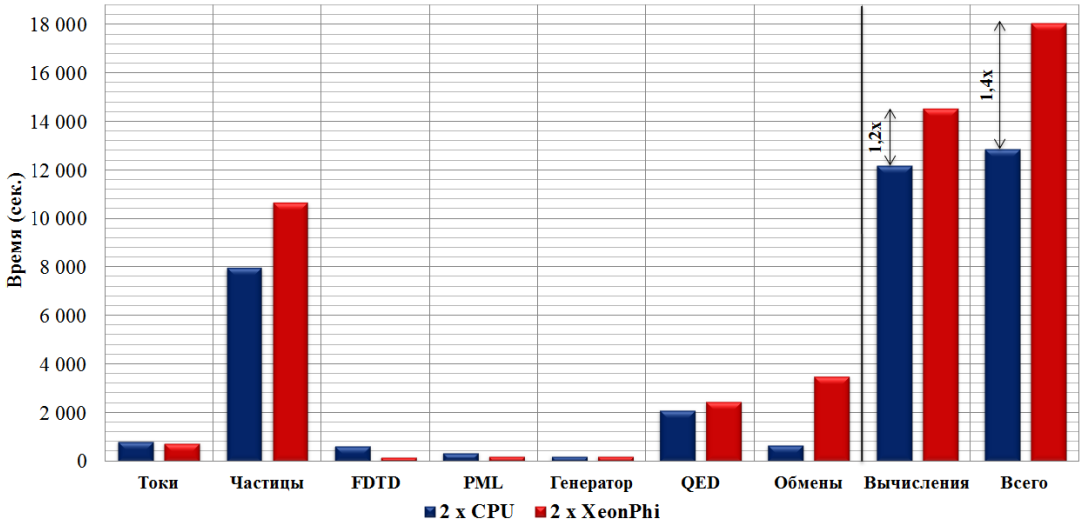

Рис. 5. Время расчета на разных этапах метода частиц в ячейках с использованием двух CPU или двух Xeon Phi. Число макрочастиц на ячейку ***ppc*** в начале расчета равно 20

Обратив внимание на отставание по времени на всех наиболее трудоемких этапах, мы изучили, как меняется соотношение времени работы вычислительной части в ходе расчета (рисунок 6). Построенная диаграмма показывает, что в начале расчета Xeon Phi демонстрирует преимущество над CPU. Далее ситуация кардинально меняется и Xeon Phi начинает существенно отставать. На заключительном этапе ситуация с соотношением времени качественно близка к начальному периоду. Построенный график изменения среднего числа части во времени (сиреневая линия) проясняет причины такого поведения. Выясняется, что в интересующий нас период времени происходят резкие скачки среднего числа макрочастиц на ячейку.

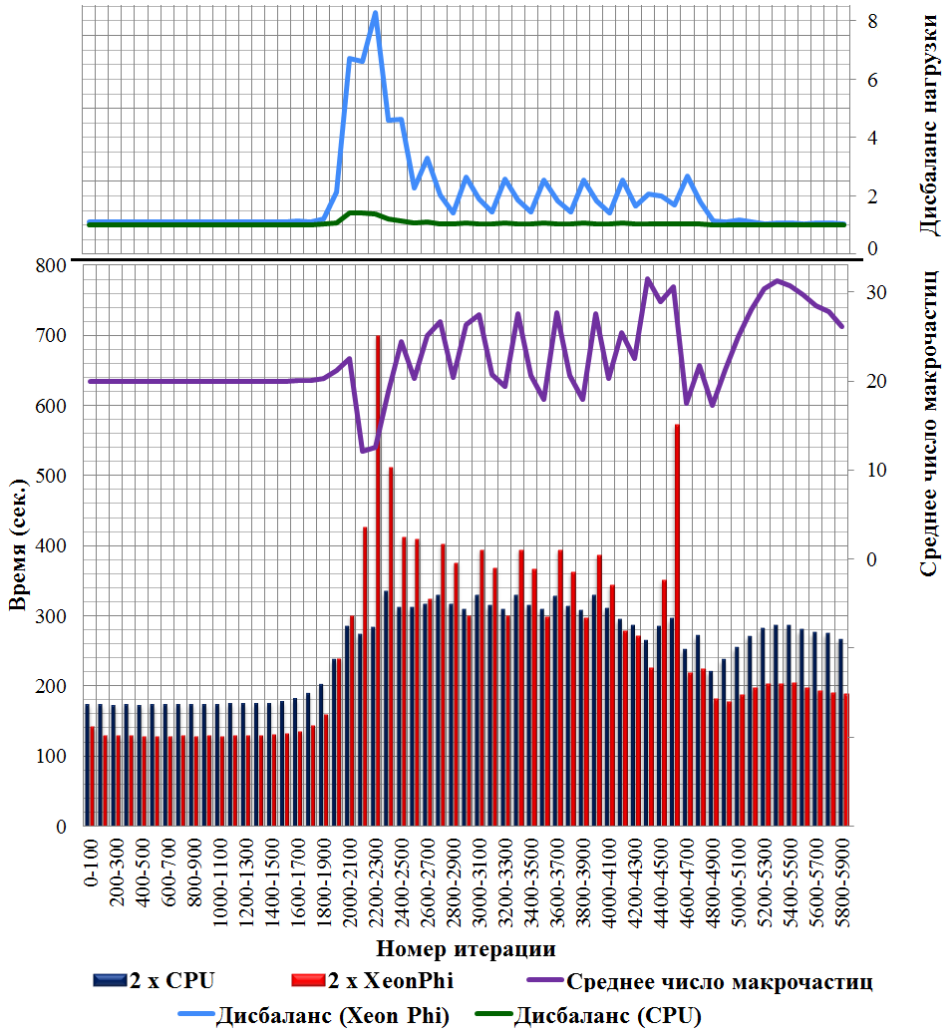

Рис. 6. Зависимость времени выполнения блоков последовательных итераций по времени от среднего числа макрочастиц на ячейку и дисбаланса нагрузки. Используются два CPU или два Xeon Phi

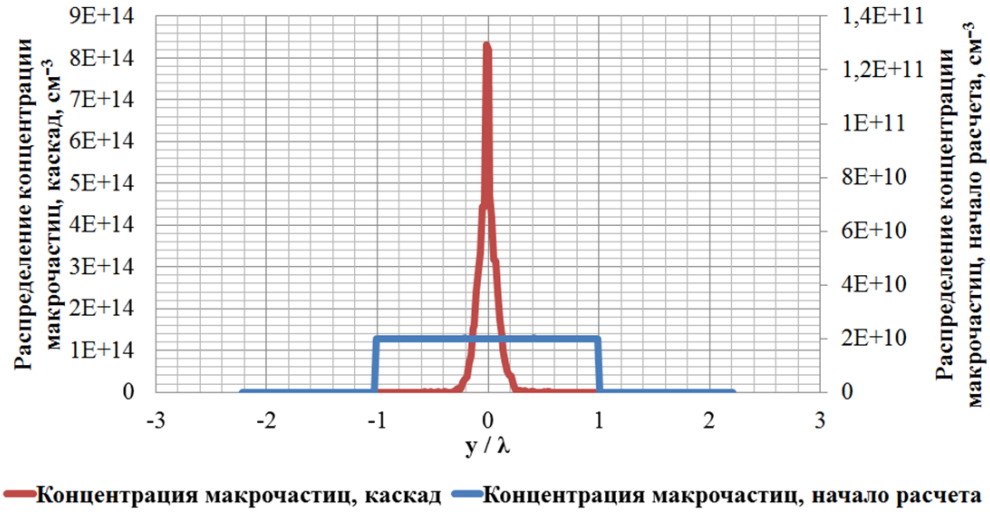

Рис. 7. Концентрация макрочастиц в начале расчета и в период возникновения каскада

Вследствие того, что преобладающее число макрочастиц сосредоточено в малой части расчетной области (рисунок 7), при вычислениях на Xeon Phi возникает дисбаланс нагрузки между потоками (голубая линия на рисунке 6). При этом отношение между максимальным и средним числом макрочастиц, обрабатываемых потоками, достигает 8. Отметим, что при аналогичном

расчете на CPU используется всего 16 потоков, в связи с чем аналогичное отношение не превышает 1,5. Наличие дисбаланса нагрузки наряду с интенсивной работой процедур создания/объединения частиц приводит к росту времени расчета как на Xeon Phi, так и на CPU. При этом потеря эффективности параллелизма в текущей реализации на Xeon Phi приводит к значительно более существенному замедлению, что и оказывает определяющее воздействие на итоговое соотношение времен. Данный эффект может быть преодолен за счет создания интеллектуального планировщика, учитывающего динамику изменения числа макрочастиц в ячейках, что является одним из перспективных направлений дальнейшей работы.

9. Заключение

В работе представлена оптимизированная для Intel Xeon Phi реализация метода частиц в ячейках для моделирования лазерной плазмы, выполненная в рамках программного комплекса PICADOR. Сформулированы основные факторы, влияющие на производительность PIC-кодов на разных уровнях параллельных вычислительных систем. В отличие от ранее опубликованных результатов успешного использования Xeon Phi при решении модельной задачи, выполнен анализ производительности в сравнении с многоядерным CPU в трех реальных расчетах.

Во всех рассмотренных задачах встречается *существенно неравномерное распределение частиц в расчетной области*. Данная разновидность *дисбаланса* может затруднить достижение хорошей производительности и на CPU, и на Xeon Phi, однако при надлежащем разбиении области на домены во всех задачах удается достичь достаточного баланса нагрузки между процессами. Следующим фактором, существенно влияющим на эффективность использования Xeon Phi по сравнению с многоядерным CPU, является число макрочастиц на ячейку, используемое в расчетах. Показано, что даже при 10 макрочастицах на ячейку Xeon Phi вполне может обогнать CPU, и эта разница увеличивается вместе с числом используемых макрочастиц.

Тем не менее, существует еще одна особенность, ограничивающая класс задач, для решения которых целесообразно использовать обсуждаемую реализацию метода частиц в ячейках для Xeon Phi. Данная особенность в полной мере проявляется в третьей задаче и приводит к отставанию по сравнению с CPU. Суть эффекта заключается в необходимости учета при моделировании процессов создания/объединения частиц в сочетании с существенным изменением их количества и положения в расчетной области. Разработка планировщика, регулирующего нагрузку между потоками на Xeon Phi, является одним из направлений дальнейшей работы.

Подводя итоги, необходимо отметить, что Xeon Phi может быть эффективно использован для численного решения задач физики плазмы программным комплексом PICADOR с учетом описанных выше ограничений.

Литература

1. Бэдсел Ч., Ленгдон А. Физика плазмы и численное моделирование: Пер. с англ. М.: Энергоатомиздат, 1989. 452 с.
2. Hockney R., Eastwood J. Computer Simulation Using Particles. IOP, Bristol and NY, 1989.
3. Tskhakaya D. The Particle-in-Cell Method // Computational Many-Particle Physics. Lecture Notes in Physics, 2008. Vol. 739. P. 161–189.
4. Tarakanov V.P. User’s Manual for Code KARAT. Springfield, VA: Berkeley Research Associates, 1992.
5. Romanov D.V., et al. Self-Organization of a Plasma due to 3D Evolution of the Weibel Instability // Phys. Rev. Lett. 2004. Vol. 93, No. 215004.
6. Popov K.I., Bychenkov V.Yu., et al. Vacuum electron acceleration by tightly focused laser pulses with nanoscale targets // Phys. Plasmas. 2009. Vol. 16., No. 053106.
7. Nerush E.N., Kostyukov I.Yu. Carrier-Envelope Phase Effects in Plasma-Based Electron Acceleration with Few-Cycle Laser Pulses // Phys. Rev. Lett. 2009. Vol. 103. No. 035001.

8. Kraeva M.A., Malyshkin V.E. Assembly Technology for Parallel Realization of Numerical Models on MIMD-Multicomputers // FGCS. 2001. Vol. 17, No. 6. P. 755–765.
9. Берендеев Е.А., и др. Моделирование на суперЭВМ динамики плазменных электронов в ловушке с инверсными магнитными пробками и мультипольными магнитными стенками // Вычислительные методы и программирование. 2013. Т. 14. С. 149–154.
10. Перепёлкина А.Ю., Левченко В.Д., Горячев И.А. Трехмерный кинетический код CFHall для моделирования замагниченной плазмы // Матем. Моделир. 2013. Т. 25, № 11. С. 98–110.
11. Fonseca R.A., et al. Exploiting multi-scale parallelism for large scale numerical modelling of laser wakefield accelerators // Plasma Physics and Controlled Fusion. 2013. Vol. 55, No. 124011.
12. Pukhov A. Three-Dimensional Electromagnetic Relativistic Particle-in-Cell code VLPL // Journal of Plasma Physics. 1999. Vol. 61. P. 425–433.
13. Bowers K.J., et al. Advances in petascale kinetic plasma simulation with VPIC and Roadrunner // J. Phys.: Conf. Ser. 2009. Vol. 180. P. 1–10.
14. Burau H., Widera R., Honig W., et al. PIConGPU: A Fully Relativistic Particle-in-Cell Code for a GPU Cluster // IEEE Transactions on Plasma Science. 2010. Vol. 33. P. 2831–2839.
15. Bastrakov S., et al. Particle-in-cell plasma simulation on heterogeneous cluster systems // Journal of Computational Science. 2012. Vol. 3. P. 474–479.
16. Bastrakov S., et al. Particle-in-Cell Plasma Simulation on CPUs, GPUs and Xeon Phi Coprocessors // ISC, LNCS. 2014. Vol. 8488. P. 513–514.
17. Бастраков С.И. и др. Динамическая балансировка в коде PICADOR для моделирования плазмы // Вычислительные методы и программирование. 2013. Т. 14. С. 67–74.
18. Сурмин И.А. и др. Моделирование плазмы методом частиц в ячейках с использованием сопроцессоров Intel Xeon Phi // Вычислительные методы и программирование. 2014. Т. 15. С. 530–536.
19. Gonoskov A., et al. Extending PIC schemes for the study of physics in ultra-strong laser fields // arXiv preprint arXiv:1412.6426.
20. Surmin I.A. et al. Particle-in-Cell Laser-Plasma Simulation on Xeon Phi Coprocessors // arXiv preprint arXiv: 1505.07271.
21. Nakashima H. Manycore challenge in particle-in-cell simulation: How to exploit 1 TFlops peak performance for simulation codes with irregular computation // Comp. & El. Engineering. 2015.
22. Bussmann M. et al. Radiative Signatures of the Relativistic Kelvin-Helmholtz Instability // Proceedings SC13: Int. Conference for HPC, Networking, Storage and Analysis, 2013. – 5-1.
23. Decyk V.K., Singh T.V. Particle-in-cell algorithms for emerging computer architectures // Computer Physics Communications. 2014. Vol. 185, No. 3. P. 708–719.
24. Taflove A. Computational Electrodynamics: The Finite-Difference Time-Domain Method. London: Artech House, 1995. 599 p.
25. Berenger J.-P. A Perfectly Matched Layer for the Absorption of Electromagnetic Waves // Journal of Computational Physics, 1994. Vol. 114. P. 185–200.
26. Esirkepov T. Exact charge conservation scheme for Particle-in-Cell simulation with an arbitrary form-factor // Computer Physics Communications, 2001. Vol. 135. P. 144–153.
27. Villasenor J., Buneman O. Rigorous charge conservation for local electromagnetic field solvers // Computer Physics Communications. 1992. Vol. 69, No. 2. P. 306–316.
28. Macchi A., Borghesi M., Passoni M. Ion acceleration by super-intense laser-plasma interaction // Reviews of Modern Physics. 2013. Vol. 85. P. 58.

29. Wilks S.C., et al. Energetic proton generation in ultra-intense laser-solid interactions // Physics of Plasmas. 2001. Vol. 8, No. 542549.

30. Green J.S., et al. High efficiency proton beam generation through target thickness control in femtosecond laser-plasma interactions // Appl. Phys. Letters. 2014. Vol. 104, No. 214101.

31. Pae K.H., et al. Proposed hole-target for improving maximum proton energy driven by a short intense laser pulse // Physics of Plasmas. 2009. Vol. 16, No. 073106.

32. Sung J. H. *et al.* // Opt. Lett. 2010. Vol. 35, No. 3021.

33. Pipahl A., Anashkina E. A., Toncian M. et al. // Phys. Rev. E. 2013. Vol. 87., No. 033104.

34. Абрамян Л.А., Литвак А.Г., Миронов В.А., Сергеев А.М. // ЖЭТФ. 1992. Т. 75. С. 978.

35. Bell A.R., Kirk J.G. Possibility of Prolific Pair Production with High-Power Lasers // Phys. Rev. Lett. 2008. Vol. 101, No. 200403.

36. Bashinov A.V., et al. New horizons for extreme light physics with mega-science project XCELS // Eur. Phys. J. Special Topics. 2014. Vol. 223. P. 1105–1112.

37. Gonoskov A., et al. Anomalous Radiative Trapping in Laser Fields of Extreme Intensity // Phys. Rev. Lett. 2014. Vol. 113, No. 014801.

Case study of using Intel Xeon Phi for solving particle-in-cell plasma simulation problems

Iosif Meyerov, Sergey Bastrakov, Igor Surmin, Arkady Gonoskov, Evgeny Efimenko, Aleksei Bashinov, Artem Korzhimanov, Anton Larin, Alexander Muraviev, Anatoly Rozanov and Mikhail Savichev

Keywords: Plasma simulation, Particle-In-Cell, Intel Xeon Phi, Case study, Performance optimization, High Performance Computing, Parallel programming, Optimization techniques

This paper considers efficient utilization of computational systems equipped with Intel Xeon Phi coprocessors for laser-plasma simulation. We analyze features of Xeon Phi architecture that influence performance of Particle-in-Cell plasma simulation. Description of the parallel plasma simulation code PICADOR previously optimized for Xeon Phi is provided. We study performance on Xeon Phi compared to CPU on three computationally intensive plasma simulation problems. A ratio of computational time on Xeon Phi to CPU is given for the main stages of the Particle-in-Cell method. Our study shows that depending on features of the physical problem using Xeon Phi can be both advantageous and disadvantageous compared to CPU.