

АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ И МАСШТАБИРУЕМОСТИ КОДА PICADOR ДЛЯ МОДЕЛИРОВАНИЯ ПЛАЗМЫ

*С.И. Бастраков¹, А.А. Гоносков², Р.В. Донченко¹, Е.С. Ефименко², А.С. Малышев¹,
И.Б. Мееров¹, И.А. Сурмин¹*

¹*Нижегородский госуниверситет им. Н.И. Лобачевского*

²*Институт прикладной физики РАН*

Введение

Теоретические исследования поведения вещества в экстремальных условиях, при которых происходит ионизация вещества, т.е. переход в состояние плазмы, приобретают всё большее значение для решения многих прикладных и фундаментальных задач. Эти исследования представляют интерес для целого ряда приложений, среди которых можно выделить лазерное ускорение заряженных частиц, концепцию быстрого поджига для управляемого термоядерного синтеза и генерацию излучения с уникальными характеристиками в труднодоступных частотных диапазонах. Среди медицинских приложений стоит отметить адронную терапию, которую используют для лечения онкологических заболеваний.

Часто, в связи с высокой степенью нелинейности и геометрической сложностью задачи, исследование динамики плазменных структур основывается на моделировании плазмы методом частиц в ячейках (particle-in-cell, PIC) [12]. Основная идея метода заключается в одновременной обработке принципиально разнородных массивов данных, содержащих информацию о координатах и скоростях заряженных частиц плазмы и об электромагнитном поле, заданном в узлах дискретной решетки, представляющей часть трехмерного пространства. Для решения прикладных задач необходимо создание специализированного программного обеспечения, ориентированного на использование суперкомпьютеров – существуют задачи, требующие моделирования $\sim 10^9$ частиц и $\sim 10^8$ узлов пространственной сетки.

В настоящее время разрабатывается ряд пакетов (PIC-кодов), основанных на методе частиц в ячейках:

- VLPL [7], разрабатываемый в немецком институте Max-Planck-Institut für Quantenoptik, первая система, позволяющая проводить эксперименты по полномасштабному трёхмерному моделированию релятивистских плазменных взаимодействий. Использует MPI для работы на кластерных системах.
- OSIRIS [4], разрабатываемый португальским институтом Instituto Superior Técnico совместно с другими институтами Европы и Америки, поддерживает двумерное и трёхмерное моделирование и, подобно VLPL, предназначен для работы на традиционных кластерных системах.
- QuickPIC [6], разрабатываемый четырьмя институтами США и Португалии, использует упрощенную модель для достижения большей эффективности вычислений, а именно стократного или большего ускорения по сравнению с OSIRIS.
- VPIC [3], разрабатываемый в лаборатории Los Alamos National Laboratory (США), который, помимо обычных кластерных систем на базе процессорной архитектуры x86, был портирован на кластеры на базе процессоров IBM Cell, чтобы обеспечить возможность запуска на суперкомпьютере Roadrunner.

- PIConGPU [5], разрабатываемый в немецком университете Technische Universität Dresden, который предназначен для работы на кластерах из графических процессоров и имеет версии для интерфейсов NVIDIA CUDA и OpenCL.

Пакет Picador [1, 2, 10, 11] разрабатывается коллективом сотрудников ННГУ и ИПФ РАН с 2010 года. Отличительной особенностью пакета является поддержка гетерогенных кластеров. Пакет позволяет выполнять расчеты как на обычных, так и на графических процессорах. В работе приводится анализ производительности и масштабируемости пакета Picador при решении модельных задач.

1. Постановка задачи и метод решения

С физической точки зрения задача моделирования плазмы заключается в моделировании пространственно-временной динамики заряженных частиц (электронов и ионов) и электромагнитного поля. Основная идея метода частиц в ячейках [12] заключается в том, что ансамбль заряженных частиц заменяется сравнительно небольшим количеством более массивных и обладающих большим зарядом частиц, с сохранением общей массы и заряда. Это упрощение позволяет моделировать движение каждой такой частицы в электромагнитном поле под действием силы Лоренца. Эволюция электрического и магнитного полей описывается при помощи уравнений Максвелла, в которые входят токи, создаваемые движением заряженных частиц.

Классическая схема метода частиц в ячейках состоит из 4 этапов: интегрирование уравнений поля, взвешивание полей, интегрирование уравнений движения, взвешивание токов. Для интегрирования уравнений поля используется алгоритм FDTD [9], [8]. На пространство накладывается равномерная сетка, внутри каждой ячейки компоненты электрического поля определяются на серединах ребер, а компоненты магнитного поля – в центрах граней. В качестве источников в уравнения Максвелла входят плазменные токи, возникающие вследствие движения заряженных частиц. В отличие от полей и токов, определенных на сетке, координата частицы является непрерывной, поэтому на этапе взвешивания полей происходит вычисление электрического и магнитного полей в точке нахождения частицы по рассчитанным ранее сеточным значениям полей. Эти значения используются при интегрировании уравнений движения с использованием алгоритма Boris [12] для вычисления положения и скорости частицы в следующий момент времени. На этапе взвешивания токов определяются сеточные значения плотности тока, возникающего вследствие движения заряженных частиц.

2. Программная реализация

Пакет Picador разрабатывается на языке C++. Вычислительно трудоемкие расчетные модули реализуются на языке C. Параллельная реализация для кластерных систем основана на использовании MPI. Используется пространственная декомпозиция: расчетная область делится на подобласти (домены) с перекрытием на границах. Операции над доменами выполняются параллельно. Расчеты в рамках каждого домена могут выполняться как на CPU, так и на GPU. Распараллеливание внутри домена основано на OpenMP (реализация для CPU) или OpenCL (реализация для GPU). Операции с плавающей запятой на CPU выполняются с использованием набора инструкций SSE. Возможно одновременное использование центральных и графических процессоров для расчетов. Более подробно с параллельной реализацией можно ознакомиться в работах [10], [11], [1] и [2].

3. Анализ производительности

Производительность прикладного программного обеспечения (ПО) зависит от многих факторов, среди которых качество кода, используемый компилятор и другое системное ПО, а также вычислительная система. В данной работе вычисляется степень

утилизации вычислительной мощности используемого процессора (как CPU, так и GPU) при работе пакета Picador. Степень утилизации можно определить, разделив число осуществлённых процессором арифметических операций с плавающей запятой (FLOP) за нужный промежуток времени (фаза расчетов) на максимально возможное число таких операций для конкретного процессора.

Чтобы замерить число реально выполнявшихся операций для CPU-версии, использовались аппаратные счётчики числа SSE-инструкций с плавающей запятой, доступные на процессорах Intel (SSEX_UOPS_RETIRED для типов PACKED_DOUBLE, PACKED_SINGLE, SCALAR_DOUBLE, SCALAR_SINGLE), и программный инструмент Intel VTune Amplifier XE. На GPU, к которому имелся доступ, нет счётчиков, позволяющих установить число арифметических операций с плавающей запятой. Поэтому для GPU-версии была получена лишь приблизительная оценка путём подсчёта арифметических операций в исходном коде и экстраполяции, основанной на параметрах задачи.

Для эксперимента использовался один из узлов кластера ННГУ (процессор Intel Xeon L5630 с частотой 2,13 ГГц). Пиковая производительность одного ядра составляет 8,52 GFLOPS для двойной точности и 17,04 GFLOPS – для одинарной. Для GPU NVIDIA Tesla X2070 (14 мультипроцессоров, частота 1,15 ГГц), установленных на этом кластере, пиковая производительность для двойной точности равна 515,2 GFLOPS, для одинарной – 1030,4 GFLOPS.

Использовался тест с ленгмюровскими колебаниями плазмы, в котором задействовано 32 768 ячеек и 983 000 частиц.

Таблица 1. Времена работы и метрики производительности для CPU и GPU версий

Метрика	Взвешивание токов	Интегрирование уравнений Максвелла	Интегрирование уравнений движения	Итого
CPU				
Выполнено операций, млрд.	97,14	3,32	444,98	545,44
Время, сек	35,052	2,635	343,784	381,471
Производительность, GFLOPS	2,77	1,26	1,29	1,43
Процент от пика	32,53%	14,77%	15,19%	16,78%
GPU				
Выполнено операций, млрд.	537,37	10,64	2127,91	2675,92
Время, сек	59,02	24,848	40,648	124,516
Производительность, GFLOPS	9,10	0,43	52,35	21,49
Процент от пика	1,77%	0,08%	10,16%	4,17%

Таким образом, в итоге мы получаем $\approx 17\%$ утилизации центрального процессора и $\approx 4\%$ – графического, что свидетельствует о наличии потенциала для оптимизации в обеих версиях. Оценка для GPU является приближённой.

4. Анализ масштабируемости

Эксперименты для оценки масштабируемости проводились на кластере Akka, который находится в High Performance Computing Center North (2x Intel Xeon L5420 2.5 ГГц, 16 ГБ RAM, Infiniband 4x, CentOS 5.6 x86_64). В качестве теста была выбрана за-

дача о ленгмюровских колебаниях плазмы, в которой 16,77 миллиона ячеек и 503,1 миллиона частиц.

Таблица 2. Результаты масштабируемости MPI-версии

Число процессов	512	1024	2048	4096
Вычисления, сек				
Токи	344,10	171,97	76,53	27,93
Поля	5,25	2,65	1,38	0,67
Частицы	687,81	345,76	170,16	83,98
(итого)	1037,16	520,38	248,07	112,58
Передачи, сек				
Токи	52,27	17,55	26,98	70,44
Поля	28,98	28,24	33,74	154,38
Частицы	52,02	36,77	36,34	112,59
(итого)	133,27	82,56	97,06	337,41
Полное время, сек	1170,43	602,94	345,13	449,99
Эффективность относительно 512 процессов	100%	96,85%	84,78%	28,56%

Плохой результат для 4096 ядер может быть объяснен тем, что тестовая задача слишком мала для такого числа процессов. Время передач для токов и полей прямо пропорционально сумме площадей граней всех доменов, то есть зависит от их размеров и количества. Для 4096 процессов домены стали слишком малыми и накладные расходы на синхронизацию передач стали занимать существенную часть времени работы.

Эксперименты, в которых одна и та же задача решается всё большим числом процессов, не вполне точно показывают реальную масштабируемость комплекса. На практике дополнительные вычислительные ресурсы можно использовать не только для решения имеющихся проблем за меньшее время, но и для решения более сложных проблем за то же время. Поэтому была проведена серия экспериментов для оценки слабой масштабируемости – способности комплекса сохранять время работы при увеличении сложности задачи пропорционально числу используемых процессов. В следующей таблице представлены результаты одного такого эксперимента на кластере Акка.

Таблица 3. Результаты слабой масштабируемости для MPI-реализации

Процессов	512	768	1024	1280	1792	2048	3072
Вычисления, сек							
Токи	161,68	162,01	160,9	162,72	161,01	160,74	160,66
Поля	4,49	4,35	4,40	4,45	4,36	4,47	4,42
Частицы	446,93	453,58	452,29	456,41	456,19	448,28	450,45
(итого)	613,10	619,94	617,59	623,58	621,56	613,49	615,53
Передачи, сек							
Токи	40,00	22,08	44,82	56,79	46,25	38,86	48,32
Поля	24,50	49,38	32,21	65,35	50,53	47,15	36,92
Частицы	33,43	41,51	62,20	54,18	70,04	75,00	80,44
(итого)	97,93	112,97	139,23	176,32	166,82	161,01	165,68
Полное время	711,03	732,91	756,82	799,90	788,38	774,50	781,21
Эффективность	1	0,97	0,94	0,89	0,90	0,92	0,91

Под эффективностью в данном случае понимается масштабированная эффективность, т.е. величина $E(p) = \frac{T(p,1)}{pT(p,p)} = \frac{T(1,1)}{T(p,p)}$, где $T(n,p)$ – время работы p процессов на

задаче размера n . Как и в предыдущем случае, мы считаем относительную эффективность, принимая эффективность запуска на 512 процессах за единицу. Слабая масштабируемость показывает хорошие результаты даже на 3072 процессах.

Эксперименты для графических процессоров были проведены на кластере ННГУ (2x Intel Xeon L5630 2.13 ГГц, 24 ГБ RAM, 2x NVIDIA Tesla X2070 (по 448 CUDA-ядер), Infiniband QDR, Windows Server 2008 HPC Edition SP2 x64). В качестве тестовой задачи были выбраны ленгмюровские колебания плазмы с сеткой, содержащей 32 768 ячеек и 983 000 частиц.

Таблица 4. Результаты работы гетерогенной версии

Точность	Двойная точность				Одинарная точность				
	Число процессоров	1 ядро CPU	8 ядер CPU	1 GPU	2 GPU	1 ядро CPU	8 ядер CPU	1 GPU	2 GPU
Вычисления, сек									
Токи	438,72	90,43	59,02	32,18	347,58	52,52	25,65	15,38	
Поля	16,03	2,53	24,85	22,11	14,81	10,36	19,40	18,10	
Частицы	2160,52	325,11	40,65	23,48	1681,99	247,26	20,57	12,84	
(итого)	2615,27	418,07	124,52	77,77	2044,38	310,14	65,62	46,32	
Передачи, сек									
Токи	4,20	3,72	4,15	2,84	3,46	3,45	3,40	2,21	
Поля	3,69	2,76	5,12	3,79	3,64	12,15	4,06	2,52	
Частицы	4,20	8,98	5,50	7,61	3,19	12,90	4,74	6,79	
(итого)	12,09	15,46	14,77	14,24	10,29	28,50	12,20	11,52	
Полное время, сек	2627,36	433,53	139,29	92,01	2054,67	338,64	77,82	57,84	

Из таблицы 4 видно, что один GPU работает быстрее, чем восемь CPU-ядер, примерно в 3,11 раза в двойной точности. К сожалению, при совмещении CPU- и GPU-процессов узла не удалось добиться более высокой производительности, чем при использовании только двух имеющихся GPU. Дело в том, что, хотя в целом GPU-процессы быстрее CPU-процессов, стадия интегрирования уравнений Максвелла на них выполняется медленнее. Таким образом, независимо от того, как расчётная область разделена между CPU- и GPU-процессами, по крайней мере одна из стадий будет выполняться разное время, и часть потоков будет вынуждена ожидать завершения вычислений в другой части.

5. Заключение

Для разработанной параллельной версии кода Picador был проведён анализ производительности, масштабируемости, слабой масштабируемости в CPU- и GPU-версиях. Анализ производительности указал на наличие потенциала для программной оптимизации кода с целью повышения утилизации вычислительных устройств. Исследование масштабируемости показало, что время работы можно сократить при помощи асинхронных передач либо подбирая число процессов исходя из размеров задачи. Рассмотрение слабой масштабируемости дало представление о возможности использования кода для решения сложных задач, а анализ гетерогенной версии выявил проблему различного времени работы для разных этапов в CPU- и GPU-версиях.

Литература

1. 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. 2012. V. 3. P. 474-479.

2. Bastrakov S., Gonoskov A., Donchenko R., Efimenko E., Malyshev A., Meyerov I. Three-dimensional Particle-In-Cell Plasma Simulation on Heterogeneous Computing Systems // Twenty-Second International Conference on Numerical Simulation of Plasmas. 2011. – [http://icnsp2011.pppl.gov/abstracts/Meyerov_I_Three-dimensional%20Particl.pdf].
3. Bergen B., Albright B.J., Bowers K.J., Daughton W., et al. Petascale Plasma Physics Simulations on Roadrunner Using VPIC. 2008. – [<http://www.lanl.gov/orgs/adts/docs/5Bergen.pdf>].
4. Fonseca R.A., Silva L.O., Tsung F.S., Decyk, V.K., Lu W., et al. OSIRIS: A Three-Dimensional, Fully Relativistic Particle in Cell Code for Modeling Plasma Based Accelerators. // Proceedings of the International Conference on Computational Science. Part III. 2002. P. 342-351.
5. Hönig W., Schmitt F., Widera R., Burau H., et al. A Generic Approach for Developing Highly Scalable Particle-Mesh Codes for GPUs. 2010. – [http://saahpc.ncsa.illinois.edu/10/papers/paper_10.pdf].
6. Huang C., Decyk V.K., Zhou M., Lu W., et al. QuickPIC: a highly efficient fully parallelized PIC code for plasma-based acceleration // Journal of Physics: Conference Series. 2006. V. 46. P. 190-199.
7. Pukhov A. High Performance 3D PIC Code VLPL: Virtual Laser Plasma Lab. 1999. – [<http://www.billingpreis.mpg.de/hbp99/pu4hbp99.pdf>].
8. Taflove A. Computational Electrodynamics: The Finite-Difference Time-Domain Method. – London: Artech House, 1995. – 599 p.
9. Yee K. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media // IEEE Transactions on Antennas and Propagation. – V. 14. – P. 302-307.
10. Бастратов С.И., Гоновсков А.А., Донченко Р.В., Ефименко Е.С., Малышев А.С., Мереров И.Б. Исследование и поиск наиболее эффективных подходов к параллельному моделированию плазмы методом частиц в ячейках на кластерных системах // Параллельные вычислительные технологии (ПаВТ'2011). Труды Международной научной конференции. – Челябинск: Издательский центр ЮУрГУ, 2011. – С. 411-417. – [<http://omega.sp.susu.ac.ru/books/conference/PaVT2011/short/118.pdf>].
11. Бастратов С.И., Гоновсков А.А., Донченко Р.В., Ефименко Е.С., Малышев А.С., Мереров И.Б. Моделирование плазмы методом частиц в ячейках на гетерогенных кластерных системах // Параллельные вычислительные технологии (ПаВТ'2012): Труды Международной научной конференции. – Челябинск: Издательский центр ЮУрГУ, 2012. – С. 367-372. – [<http://pavt.susu.ru/2012/short/062.pdf>].
12. Бэдсел Ч., Ленгдон А. Физика плазмы и численное моделирование: Пер. с англ. – М.: Энергоатомиздат, 1989. – 452 с.