

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Санкт-Петербургский государственный университет

ВЫСОКОПРОЗВОДИТЕЛЬНЫЕ ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ НА КЛАСТЕРНЫХ СИСТЕМАХ

Материалы шестого Международного
научно-практического семинара

Том 2

12–17 декабря 2006 г.

Издательство Санкт-Петербургского госуниверситета
Санкт-Петербург
2007

УДК 681.3.012:51
ББК 32.973.26–018.2:22
В 93

В93 Высокопроизводительные параллельные вычисления на кластерных системах. Материалы шестого Международного научно-практического семинара. Том 2. / Под ред. проф. **Р.Г. Стронгина.** Санкт-Петербург: Изд-во Санкт-Петербургского госуниверситета, 2007. 255 с.

Сборник сформирован по итогам научного семинара, посвященного теоретической и практической проблематике параллельных вычислений, ориентированных на использование современных многопроцессорных архитектур кластерного типа. Семинар проходил в Санкт-Петербурге 12–17 декабря 2006 года.

Вошедшие в сборник материалы семинара представляют интерес для преподавателей и научных сотрудников высших учебных заведений, а также для инженеров, аспирантов и студентов вузов.

Отв. за выпуск к.ф.-м.н. **И.М. Григорьев**

ББК 32.973.26–018.2:22

Поддержка семинара



Компания Intel Technologies

© Санкт-Петербургский госуниверситет, 2007

гибридный подход на кластерах ПК, содержащих симметричные мультипроцессоры в узлах.

ПРИМЕНЕНИЕ ПРИКЛАДНЫХ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ В НАУЧНЫХ ИССЛЕДОВАНИЯХ

Митрохин Ю.С., Ковнеристый Ю.К., Шудегов В.Е.

*Удмуртский государственный университет, г. Ижевск,
Институт металлургии им. А.А.Байкова РАН, г. Москва*

Введение

Методы моделирования в физике, химии и в других областях знаний требуют очень больших вычислительных затрат и предъявляют жесткие требования к имеющейся у исследователя вычислительной технике. Раньше такие работы можно было выполнять только на мощных вычислительных центрах которые были дороги и труднодоступны. С появлением персональных компьютеров и ростом их производительности положение дел в этой области изменилось радикально. В настоящее время настольный компьютер имеет более высокую производительность чем большой компьютер 20-летней давности. А после появления параллельных кластеров стало возможным проведение серьезных численных экспериментов во многих ВУЗах и институтах. Одновременно с развитием вычислительной техники развивалось и программное обеспечение для выполнения параллельных вычислений. Основным языком программирования в задачах моделирования был и остается язык Fortran, хотя язык С тоже получает распространение в современных программах. Основным средством создания параллельных программ для распределенных кластеров является MPI. Для компьютеров с общей памятью можно использовать OpenMP. Почти все современные трансляторы с языков Fortran 90 и С имеют опции для работы с OpenMP. То есть современные трансляторы позволяют автоматизировать процесс написания параллельных программ.

В связи с этим, во многих ведущих научных центрах стали создаваться большие и сложные комплексы прикладных программ для моделирования реальных физических и химических процессов. Часть из этих программ являются коммерческими а часть свободно-распространяемыми на основании GNU лицензии. Многие пакеты программ хорошо документированы и имеют много примеров

входных и выходных файлов, что значительно облегчает их освоение. Кроме того, имеется еще возможность подписаться на mailing list. Эта возможность позволяет пользователям задавать по электронной почте вопросы разработчикам программ и получать на них квалифицированные ответы. При инсталляции программ, обычно, выдается список всех платформ на которых может быть установлен данный пакет. Наибольшее распространение в настоящее время получила платформа Intel и ОС Linux. Даже там, где есть более мощная техника, обучение пользователей проводится на платформе Intel. Это связано не только с массовым распространением этой платформы, но и с тем, что трансляторы с языка Fortran 90 для персональных компьютеров хорошо отлажены и создают работоспособные загрузочные модули. На других платформах во время инсталляции и последующей работы программ часто возникают проблемы. Хуже всего с этим обстоят дела на платформе Silicon Graphics.

Методы моделирования, теория

Наибольшее распространение при моделировании физических и химических систем получил метод молекулярной динамики (МД). Он появился вместе с первым языком высокого уровня Fortran. В следующем году мы будем отмечать 50-летний юбилей этого метода [1]. В классическом методе МД моделирование проводится на атомном уровне, но из-за большой массы атомных ядер, движение атомов можно описывать классическим уравнением Ньютона:

$$\vec{F} = m\vec{a} . \quad (1)$$

При этом, основной проблемой классической МД было и остается – выбор модели межатомного взаимодействия, т.е. выбор параметров и аналитического вида потенциала межатомного взаимодействия атомов. Например, из соотношений Коши для упругих констант в кубическом кристалле:

$$c_{12} = c_{44} \quad (2)$$

следует, что парный межатомный потенциал не может правильно описать межатомное взаимодействие атомов в переходных металлах,

т.к. там это соотношение, в основном, не выполняется. Для кремния и углерода пришлось вводить специальный более сложный 3-х частичный потенциал [2]. Для металлических систем в работе [3] (Sutton-Chen) был предложен многочастичный потенциал:

$$E_{tot} = \sum_{i=1}^N V_i = \sum_{i=1}^N \left[\frac{1}{2} \sum_{j \neq i}^N \varepsilon_{ij} \left(\frac{a}{r_{ij}} \right)^n - c_i \varepsilon_{ii} \sqrt{\rho_i} \right], \quad (3)$$

где

$$\rho_i = \sum_{j \neq i}^N \left(\frac{a}{r_{ij}} \right)^m.$$

Потенциал Саттона-Чена (3) является простейшей реализацией метода погруженного атома (EAM) [4], и в настоящее время наиболее часто применяется для моделирования металлических систем.

В пионерской работе Car Parinello [5], был предложен новый подход моделирования методом молекулярной динамики из первых принципов (ab initio МД). Основное отличие состояло в том, что силы межатомного взаимодействия находились здесь, на основании хорошо известной теоремы Геллманна-Фейнманна, как градиент от полной энергии моделируемой системы:

$$\vec{F} = - \frac{\partial E_{tot}(\rho)}{\partial \vec{R}},$$

где полная энергия E_{tot} зависит от расположения атомов в суперячейке, а следовательно, и от распределения электронной плотности ρ . Для этого нужно было решить квантово-механическую задачу для заданного распределения атомов в расчетной области, т.е. решить уравнение Кона-Шема [6] (аналог уравнения Шредингера для кристалла). Однако, после того как найдены силы, дальнейшая схема моделирования была такой же как и раньше, т.е. движение атомов по-прежнему описывалось с помощью классической механики (1). В этом методе нет подгоночных параметров, а следовательно нет неконтролируемых приближений. Предложенный новый подход получил название метода Car Parinello (CPMD). Таким именем был

назван один из известных пакетов программ разработанных в филиале фирмы IBM в Швейцарии, где работает один из авторов этого метода (M.Parinello). Хотя теорема Геллманна-Фейнманна была известна еще с 30-х годов прошлого века, ее практическое применение стало возможно только после появления мощных современных суперкомпьютеров. Дело в том, что электронная часть задачи, т.е. вычисление распределения электронной плотности в суперячейке большого размера (50-100 атомов), сама по себе, требует очень больших вычислительных затрат. В методе CPMD она решается многократно и на нее уходит больше 99% машинного времени.

Среди других пакетов *ab initio* МД можно назвать следующие: VASP, CASTEP (коммерческие), ABINIT, PWSCF, SIESTA. В этих пакетах число атомов в моделируемой системе, обычно, не превышает 100. Очевидно, этого не достаточно для получения хорошей статистики. Также мал и временной интервал моделирования, так как обычно, число временных шагов не превышает 2000-3000, а величина шага составляет 1–3 fs (10^{-15} сек.). Но даже для такого малого числа атомов и временных шагов, компьютерное время моделирования на современных параллельных кластерных системах измеряется сотнями часов. Для сравнения можно сказать, что в классической МД число атомов в расчетной области составляет 10^3 - 10^6 , а число временных шагов часто превышает 10^6 .

Таким образом, можно сделать вывод, что с появлением *ab initio* МД, классическая молекулярная механика не утратила своих позиций. Для сложных органических молекул типа ДНК, она по-прежнему остается единственным инструментом для моделирования. Эти два подхода взаимно дополняют друг друга. Если раньше параметры потенциала межатомного взаимодействия можно было получить только из эксперимента методами многомерной аппроксимации, то теперь это можно сделать на основании первопринципных расчетов. Такой подход был продемонстрирован в работе [7]. Автор этой работы использовал метод погруженного атома (EAM). Параметры EAM он получил на основании первопринципных методов моделирования для той же системы. При этом полученные им результаты очень хорошо совпадали с экспериментом. Среди пакетов реализующих классическую МД можно назвать MOLDY и DL_POLY [8].

Моделирование системы Ni₃Al, результаты

На 12-процессорном кластере PARC в УдГУ были выполнены работы по моделированию двойных и тройных сплавов на основе Ni₃Al в твердом и жидком состоянии методами классической и первопринципной молекулярной динамики. Использовались как классические пакеты МД (DL_POLY) [8], так и первопринципные VASP [9-11]. Кроме того, электронная структура этих сплавов рассчитывалась зонными методами TB-LMTO-ASA [12] FP-LMTO[13]. При моделировании пакетом VASP, число атомов в суперячейке было равно 64. Исходная конфигурация атомов соответствовала гранецентрированной кубической решетке (ГЦК). Шаг интегрирования по времени был выбран равным 1.5 fs (10^{-15} сек), число временных шагов составляло 3000. Полное время моделирования одного варианта при фиксированной температуре T, составляло более 100 часов. Максимальная производительность достигалась на 6 процессорах.

Такой же расчет был выполнен на 2-х процессорном кластере Opteron в ИМЕТ. Там полное расчетное время для одной точки по температуре было равно 16 часов. Для сравнения, этот же расчет был повторен на 2-х процессорном кластере Itanium на ВЦ РАН (г.Москва). Время расчета уменьшилось в 3 раза и составило немногим более 5 часов. Если исходить из сравнения цена/стоимость, то можно сделать вывод, что хотя Itanium считает в 3 раза быстрее, стоимость его тоже примерно в 3 раза больше. По этой причине многие пользователи выбирают Opteron. Мы тоже сделали такой выбор. Здесь следует отметить один важный для практики нюанс. На Opteron'е у нас было 4Гб оперативной памяти, а на Itanium – 16Гб. При работе в параллельном режиме почти все параллельные программы завершаются аварийно при превышении оперативной памяти. Поэтому важно иметь не только мощный компьютер, но и соответствующую ему оперативную память. На Itanium все расчеты прошли без сбоев, а на Opteron'е пришлось уменьшить потребности а ОП до 4Гб.

При моделировании классической МД (пакет DL_POLY) использовались следующие параметры: число атомов в суперячейке было равно 1372 (5-кратное размножение элементарной ячейки по каждой из осей x,y,z), временной интервал был тот же 1.5 fs, число шагов на стадии релаксации – 300000, число шагов на стадии сбора статистики – 1000000. Время моделирования одного варианта по температуре на кластере PARC составляло 25 часов. Для получения

загрузочных модулей использовался транслятор IFC-9.0 фирмы Intel совместно с библиотекой MKL-8. Библиотека MKL дает выигрыш на процессорах Intel до 20-30 %. Кроме того, она полностью соответствует транслятору IFC, что снимает многие проблемы при сборке загрузочных модулей. Все расчеты проводились в OS Linux (SuSE 9.0).

В данной работе моделировался процесс плавления сплава Ni_3Al . Для этого использовался ансамбль NPT в пакете DL_POLY и ансамбль NVT в пакете VASP. В обоих случаях использовалось одно и то же уравнение состояния, которое бралось из результатов моделирования в NPT ансамбле. Температурный интервал составлял 300 К – 2000 К с шагом 100 К. При каждой температуре, начиная с 300 К система сначала подвергалась релаксации ($3 \cdot 10^5$ МД шагов), а затем производилось усреднение и расчет физических характеристик системы. Основные характеристики – это полная энергия системы, коэффициент диффузии и радиальная функции распределения атомов $g(r)$. Точка плавления определялась по скачку на графике зависимости полной энергии системы от температуры $E(T)$, по резкому увеличению коэффициента диффузии D и по изменению формы кривой функции радиального распределения атомов $g(r)$.

На рис. 1 приведены радиальные функции распределения $g(r)$ для пары атомов Ni-Ni для четырех состояний моделируемой системы с температурами $T = 300$ К, $T = 1300$ К, $T = 1600$ К и $T = 1700$ К, слева направо и сверху вниз, соответственно. Температура $T = 300$ К соответствует комнатной температуре и $g(r)$ здесь имеет форму характерную для кристаллического состояния. Следующая кривая $g(r)$ для $T = 1300$ К соответствует сильно нагретому образцу. Здесь мы видим, что начинается тепловое размытие пиков $g(r)$, т.е. происходит частичное разрушение кристаллической решетки. Третий график, $T = 1600$ К, соответствует состоянию предплавления. Кристаллическая решетка почти разрушена, но остались еще кластеры с кристаллической структурой. И на последнем графике мы имеем $g(r)$ типичную для жидкого состояния. Таким образом, можно сделать вывод, что в нашем численном эксперименте температура плавления сплава Ni_3Al равна 1700 К. Экспериментальное значение температуры плавления для этого сплава – 1650 К [16]. Температура плавления чистого никеля равна 1726 К.

Здесь нужно отметить, что точность определения температуры в таком численном эксперименте невысокая: $\Delta T = \pm 100$ градусов.

Основное его назначение – проверка модели межатомного взаимодействия и надежности программного обеспечения. Это очень жесткий тест для любой программы молекулярной динамики. Для сравнения нами был выполнен аналогичный расчет этой же системы с парным потенциалом Морзе [13, 14]. В этом случае точка плавления получилась на 300 градусов выше, что говорит о том, что парный потенциал Морзе плохо описывает эту систему. Этого и следовало ожидать, так как соотношения Коши (2) хуже всего выполняется для никеля в ряду 3d- переходных металлов. Если же не учитывать уравнение состояния ($V = \text{const}$), или неправильно его учитывать, то плавление может не быть, даже при значительном перегреве системы.

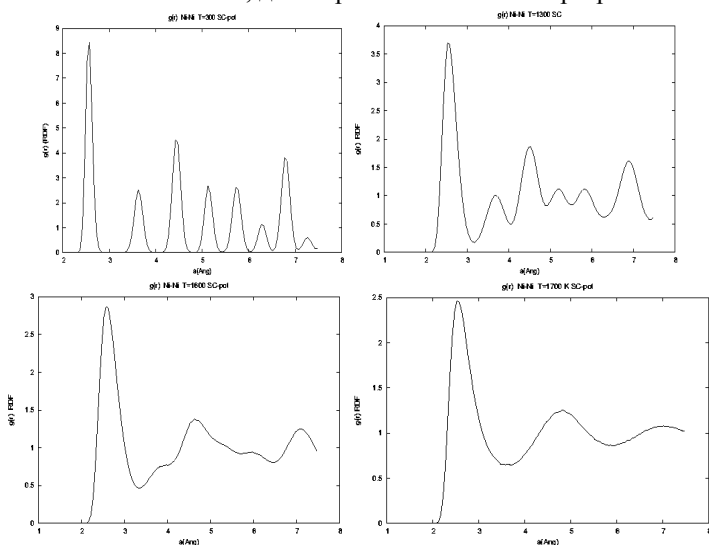


Рис. 1

Литература

1. B.J.Alder, T.E.Wainwright, J. Chem. Phys., 27, 1208 (1957).
2. J.Tersoff, Phys. Rev. B, 37, 6991 (1988).
3. A.P.Sutton, J.Chen, Phil. Mag., 61, 139 (1990).
4. M.S.Daw, M.I.Baskes, Phys. Rev. B, 29, 6443 (1984).
5. R.Car, M.Parinello, Phys. Rev. Lett., 55, 2471 (1985).
6. W.Kohn, L.J.Sham, Phys. Rev. 140, A1133 (1965).
7. F.Ercolessi, Europhys. Lett., 26, 583 (1994).
8. W.Smith, T.Forester, J.Molec. Graphics, 14, 136 (1996).

9. G.Kresse, Hafner, Phys. Rev. B, 47, RC558 (1993).
10. G.Kresse, J.Futhmuller, Comp. Mat. Sci., 6, 15 (1996).
11. G.Kresse, J.Furthmuller, Phys. Rev. B, 54, 11169 (1996).
12. O.K.Andersen, Phys. Rev. B, 12, 3060 (1975).
13. S.Y.Savrasov, Phys. Rev. B, 54, 16470 (1996).
14. P.M.Morse, Phys. Rev., 34, 57 (1929).
15. L.A.Girifalko, V.G.Weizer, Phys. Rev., 114, 687 (1959).
16. Н.Н.Степанова и др., ФММ, 1999, Т. 88, Вып. 4, С. 69-75.

Контакты

Тел. +7 (3412) 916-089, 916-090

e-mail: mit@uni.udm.ru

Трёхуровневая система разделения кода для разработки высокопроизводительных параллелизуемых программ

Монахов В.В., Керницкий И.Б., Евстигнеев Л.А.

СПбГУ, Санкт-Петербург

Введение

Одной из важных особенностей современного программного обеспечения является сочетание использования высокой производительности компьютеров с развитыми средствами взаимодействия с пользователем. При этом возникает несколько проблем. Во-первых, большинство библиотек для высокопроизводительных вычислений в области физики написано на языке FORTRAN, что не даёт возможности пользователям использовать другие языки программирования, а также создавать программы с развитым пользовательским интерфейсом и обеспечивать поддержку системного программирования. Во-вторых, “распараллеливание” программы, первоначально рассчитанной на работу в обычном непараллельном режиме, требует больших затрат. В-третьих, программы, разработанные на персональных компьютерах, часто плохо переносимы в распределённые среды. Нами предлагается стратегия решения данных проблем путём разделения исходного кода программы на три уровня.