

использование для определения реактивности уравнений (36), (37) позволяет корректно ее определять для всех этапов переходных процессов и вводить оценку локальных значений;

реактивность, вычисляемая по регистрируемой плотности потока нейтронов, определяется условиями регистрации и используемой математической моделью процесса, а погрешности значений связаны с интерпретацией локального изменения плотности потока нейтронов, привлечением времени жизни нейтронов в активной зоне и методами компенсации пространственных эффектов реактивности, которые не устраняются по результатам экспериментов.

Следует напомнить об отсутствии общепринятого положения о том, как определять реактивность для распределенных переходных процессов с обратными связями на всех этапах их протекания, а также в случае наличия посторонних источников нейтронов в активной зоне. Оценки эффектов реактивности в точках пространства состояний, далеких от эксплуатационной траектории, по результатам экспериментов имеют погрешности вследствие неопределенности коэффициентов реактивности и параметров поверхности $\rho(s_k)$.

Следует также отметить, что знание реактивности хотя и необходимо для представления об изменении плотности потока нейтронов, но не дает исчерпывающей информации на всем протяжении переходного процесса, так как:

на этапах возмущения и стабилизации ее значение не определяет локального изменения амплитудного множителя;

в каждой точке активной зоны необходимо учитывать вклад формфункции в описание динамики плотности потока нейтронов;

нельзя представлять пространственное перераспределение плотности потока нейтронов только через эффекты реактивности;

при скомпенсированных процессах в активной зоне значения реактивности не информативны.

Автор благодарен Ю.А. Крайнову и П.А. Болобову за содействие и критику при обсуждении работы, способствовавших ее углублению и появлению в печати.

СПИСОК ЛИТЕРАТУРЫ

1. Хетрик Д. Динамика ядерных реакторов. М.: Атомиздат, 1975.
2. Бэлл Д., Глестон С. Теория ядерных реакторов. М.: Атомиздат, 1974.
3. Динамика ядерных реакторов. Под ред. Я.В. Шевелева. М.: Энергоатомиздат, 1990.
4. Кори Г., Кори Т. Справочник по математике. М.: Наука, 1970.
5. Ionov V.S., Krainov Yu.A. Experimental investigation of the space — time kinetics effects at full scale critical facility V-1000. — In: Proc. Intern. Conf. on Reactor Physics and Reactor Computation. Tel-Aviv, January 23—26, 1994.

Поступила в Редакцию 21.04.94

УДК 621.039.51

ИСПОЛЬЗОВАНИЕ МНОГОПРОЦЕССОРНЫХ КОМПЬЮТЕРОВ В НЕЙТРОННО-ФИЗИЧЕСКИХ РАСЧЕТАХ РЕАКТОРА

Городков С.С. (ИЯР РНЦ «Курчатовский ин-т»)

Основной физической причиной аварии на Чернобыльской АЭС явились, как известно, ошибочные представления о свойствах активной зоны РБМК, основанные на упрощенных расчетах [1]. Отдавая должное профессиональным качествам людей, выполнявших расчеты, необходимо тем не менее признать значительную сложность задачи расчета физики этого реактора. Так, трехмерный двухгрупповой поядеичный

сеточный расчет с 16 слоями по высоте только одного временного (по неявной схеме) шага без теплогидравлики и учета сложной зависимости ячеечных характеристик от параметров зоны требует ~ 250 млн операций с плавающей запятой. Добавим, что производительность самой мощной в то время ЭВМ БЭСМ-6 ИАЭ им. И.В. Курчатова составляла около 0,3 млн операций с плавающей запятой в секунду (MFLOPS), и таких машин на многотысячный коллектив было всего три.

А ведь годы, предшествовавшие аварии и последовавшие за ней, за рубежом были ознаменованы замечательными достижениями в производительности вычислительных машин. Правда, эти успехи в значительной мере были достигнуты с помощью специфических средств. Так, первые суперкомпьютеры существенно использовали так называемую конвейерную обработку, которая позволила специально переработанным — векторизованным — программам считать во много раз быстрее. К сожалению, далеко не каждый алгоритм может быть векторизован, успешная векторизация программы требует высокого мастерства. В силу многих, в том числе и этих причин ЭВМ с векторными процессорами в нашей стране не получили распространения, а программа НЕМ-3 [2, 3] осталась, по-видимому, единственной высоковекторизованной реакторной программой.

Ныне для разработчиков и пользователей суперкомпьютеров ключевым словом является параллельная обработка, которая проводится на многопроцессорных ЭВМ. Успешный перевод обычных, «последовательных» программ на такие ЭВМ также сложен, об этом и пойдет речь в статье.

Но сначала определим более четко, что такое MFLOPS, число которых одно время было едва ли не важнейшей характеристикой ЭВМ, приводившейся в рекламных буклетах о суперкомпьютерах. При этом, к сожалению, оказывается, что производительность ЭВМ для практических расчетов гораздо ниже объявленной. Ближайший и не самый сильный пример — БЭСМ-6, выполнявшая будто бы 1 млн операций в секунду. Вообще скорость выполнения различных программ, т.е. различных наборов операций, при переводе их с одной ЭВМ на другую меняется по-разному. Поскольку в НЕМ-3 в основном используются последовательные операции внутри длинных циклов, мы произвольно положили, что при выполнении последовательного цикла

$$DO\ 1\ I = 1,5000$$

$$1\quad A(I) = B(I) * C + D(I)$$

проводится 10 тыс. операций с плавающей запятой. Именно о таких понятных любому пользователю операциях будет идти речь в настоящей работе, в то время как рекламная информация, вероятно, основывается на скорости сложения двух чисел из фиксированных ячеек кэш-памяти или на чем-то еще более абстрактном.

Производительность некоторых ЭВМ, оцененная по простейшей тестовой задаче, и время, необходимое для нейтронно-физического расчета РБМК на одном временном шаге, следующие:

	Производительность, MFLOPS	Расчет РБМК, с
БЭСМ-6	0,3	830
CYBER-936 (без оптимизации)	1,5	160
CYBER-936 (с оптимизацией)	3,4	74
IBM-PC/XT 286/287	0,034	7300
IBM-PC 386/387, 30 МГц	0,24	1040
IBM-PC 386/Weitek	0,56	450
IBM-PC 486/487, 66 МГц	1,68	150
Spark-10 RISC	3,56	70

Помимо различных ПЭВМ сюда включены две базовые машины. БЭСМ-6 попала сюда для обозначения масштаба перемен в вычислительных возможностях, CYBER-936 — как наиболее мощная в настоящее время в некоторых отечественных институтах, занимающихся расчетом реакторов. Предложенный нами простой тест хорошо оптимизируется при компиляции на этой ЭВМ, равно как и программа НЕМ-3. Наиболее мощной среди исследованных ЭВМ оказалась редкая пока у нас рабочая станция Spark, но даже и на ней расчет РБМК требует существенного времени.

Эти данные косвенно подтверждают, что мощность единичного процессора возрастала не быстро, а вычислительные возможности развивались в значительной мере за счет большей доступности таких процессоров. Ныне, когда число процессоров сопоставимо с числом сотрудников, такое развитие достигло определенного предела и если не остановится совсем, то по крайней мере существенно замедлится. Изменить эту тенденцию может только одновременное параллельное использование в одном расчете многих процессоров. Если не все, то большинство природных процессов являются параллельными. Моделируют же их алгоритмы в большинстве последовательные, что объясняется не столько особенностями человеческого мышления, сколько однопроцессорностью ЭВМ, доступных во время создания соответствующих алгоритмов.

Следует также отметить, что многопроцессорные ЭВМ долго пребывали в стадии разработки, т.е. были малодоступными, разнотипными, малосерийными или штучными, со слабым и узкоспециализированным матобеспечением и неясными перспективами дальнейшего использования. Все это никак не стимулировало разработку новых параллельных алгоритмов и распараллеливание старых, тем более что растущие вычислительные потребности удовлетворялись доступностью однопроцессорных ЭВМ.

Сейчас ситуация значительно изменилась. Назрели новые задачи, требующие существенно большего объема вычислений. В то же время разные фирмы выпускают сходные типовые элементы, из которых могут собираться вычислительные комплексы с числом процессоров от нескольких единиц до десятков тысяч. В качестве языка программирования может применяться стандартный Фортран с небольшими добавлениями, необходимыми для организации взаимодействия процессоров. Сейчас для отечественных программистов становятся доступными ЭВМ нового типа, и в первую очередь те из них, которые используют транспьютеры в качестве базовых элементов.

Транспьютер — это специализированный процессор с собственной памятью от 1 Мбайта, не только быстро выполняющий арифметические операции, но и адекватно быстро передающий данные по своим четырем каналам связи — линкам. На одной плате ПЭВМ можно смонтировать несколько транспьютеров, и несколько таких плат в том же объеме будут иметь вычислительную мощность, доступную не так давно только суперкомпьютерам. Линки различных транспьютеров можно соединять, образуя требуемую конфигурацию. Для этого надо задать соответствующую информацию для программы-конфигуратора, которая и установит характер связей между ними на время выполнения данной задачи.

При распараллеливании какой-либо программы на несколько транспьютеров в каждый из них заносится определенный сегмент, который может быть выполнен параллельно с другими сегментами. В начальный момент специализированная операционная система одновременно запускает их, и каждый транспьютер выполняет свой сегмент независимо от других до тех пор, пока не встретит оператор передачи (присма) какой-либо информации по одному из линков. Здесь транспьютер ждет, пока другой транспьютер, связанный с ним этим линком, не будет готов принять (передать) эту информацию. Так осуществляется синхронизация работы всех транспьютеров. Главной задачей программиста при распараллеливании программы является организация обмена информацией между транспьютерами.

Проиллюстрируем задачу на примере программы НЕМ-3, рассчитывающей нейтронное поле в больших энергетических реакторах. По-видимому, не вызовет возражения утверждение, что самым простым, а возможно, и наиболее эффективным способом распараллеливания алгоритма является тот, который естественно вытекает из его структуры. Трехмерный алгоритм программы НЕМ-3 получили в свое время путем развития двумерного. Поэтому в нем ведется однотипный расчет по горизонтальным слоям трехмерной сетки, что и предположило использование для расчета в каждом слое отдельного процессора при сохранении архитектуры программы.

В таком подходе привлекает не только экономия в объеме текста, подлежащего переработке, но и сохранение векторизованности программы. Дело в том, что расчеты в пределах одного слоя в НЕМ-3 оформлены в виде простых безусловных циклов по всем его узлам, что позволяет ускорять расчет в 5—7 раз при наличии векторного процессора ЭВМ. А это — качество, которое жалко терять.

Еще одним аргументом в пользу распараллеливания расчета по принципу слой—процессор явилось небольшое число процессоров в доступной нам ЭВМ, близкое к обычному числу слоев в трехмерных расчетах больших реакторов. Поскольку в сеточных алгоритмах слои явно взаимодействуют только с ближайшими соседями и обрабатываются по одинаковым правилам, соответствующие им процессоры должны быть соединены последовательно и содержать один и тот же набор программ.

Очевидно, что распараллеливание тем более эффективно, чем меньше вычислений в исходном алгоритме будет выполняться вне циклов по слоям. Действительно, каждый такой цикл в дальнейшем будет означать одновременную работу группы процессоров, и чем больше они будут загружены, тем быстрее кончится счет. Ясно также, что предварительная минимизация типов таких циклов сократит работу по распараллеливанию. В сеточной программе в циклах по слоям выполняются рассылка по слоям внешних параметров и подготовка данных для расчета, затем итерационный процесс решения с передачей результатов каждой итерации в соседние слои, затем окончательная обработка результатов, т.е. как минимум три различных процесса, реально же их больше. Чтобы отразить этот факт, циклы по слоям после минимизации их числа в исходном тексте были заменены вызовом подпрограммы SBHOST (Nproc), где Nproc — номер процесса. В последовательном алгоритме единственная ее задача — вызвать другую подпрограмму SBROOT (Nproc), которая в цикле по слоям выполнит указанный процесс.

В параллельном алгоритме программа SBHOST должна обеспечить одновременную работу группы процессоров, сообщив им предварительно по цепочке номер процесса, который надлежит выполнять в данный момент, и снабдив их необходимой для этого информацией. Находящаяся же в каждом из этих процессоров программа-аналог SBROOT, получив соответствующую информацию, должна передать ее следующему процессору, выполнить требуемые вычисления и обменяться результатами со своими соседями.

Описанные изменения, конечно, были не единственными. Все необходимые подпрограммы надо было подготовить для одновременных независимых вычислений в различных слоях. Потребовалось изменить способ расположения информации в их общих областях памяти, чтобы избежать ненужных трат и разместиться в пределах 1 Мбайта. Наконец, вместо одного набора подпрограмм было сделано четыре аналогичных, что позволило симитировать четырехпроцессорный расчет на обычной последовательной ЭВМ и устранить при этом большую часть неизбежных ошибок.

В результате все различия последовательного и параллельного вариантов НЕМ-3 оказались сосредоточены в двух подпрограммах, общий объем которых в первом случае 50 строк, во втором, применительно к транспьютерам, ~400, что составляет около 5% всего текста программы.

Расчеты выполнялись в Научном центре нейροкомпьютеров Российской академии наук на ЭВМ, объединяющей 25 транспьютеров Т800. Рекламная производительность каждого из них 1,5 MFLOPS. Описанный тест дал вместо этого 0,32 MFLOPS, что

сравнимо с производительностью ПЭВМ с процессором 386/387. Скорость обмена информацией между транспьютерами оценена в 1,2 Мбайта в секунду.

При идеальном распараллеливании по слоям скорость счета не зависела бы от числа слоев. В нашем случае вариант с 16 слоями выполнялся на 18% медленнее, чем с двумя слоями. Это объясняется прежде всего тем, что в многослойном расчете в отличие от двуслойного результаты итерации надо сообщать не одному, а двум соседям, а кроме того гораздо дальше засылать по цепочке процессоров исходную информацию и собирать окончательные результаты. Упомянутый расчет РБМК потребовал 63 с, что эквивалентно производительности 3,95 MFLOPS — самой большой из всех перечисленных в данной работе. Т800, как и процессор 386/387, в настоящее время изрядно устарел. У Т9000, современника нынешних RISC, объявленная мощность 25 MFLOPS. Если его реальная мощность также в 5 раз меньше, как у Т800, то на 16 Т9000 расчет РБМК займет 4 с. Такая скорость позволяет моделировать умеренно быстрые процессы в РБМК в реальном масштабе времени, гораздо более детально исследовать гипотетические аварийные ситуации или процесс пространственного выгорания, или различные переходные режимы.

Сейчас большие усилия по повышению безопасности ядерной энергетики направляются на исследования тяжелых аварий. Ни в коей мере не отрицая их полезность, выскажем все же убеждение, что основным направлением повышения безопасности АЭС должна стать гораздо более достоверная прогнозируемость их безаварийной работы во всех предполагаемых режимах. Здесь уместно напомнить, что на Чернобыльской АЭС авария произошла отнюдь не в результате проявления неизвестных законов природы. Ее возможность заведомо проявилась бы в исследованиях с помощью больших моделирующих программных комплексов, основанных на известных уже тогда физических моделях, если бы такие комплексы были и могли работать достаточно быстро для широкого анализа вариантов.

Высококачественные прогнозы могут быть обеспечены только при использовании сложных моделирующих комплексов, способных выполнять огромный объем вычислений в режиме реального или даже ускоренного времени. Без широкого использования параллельных вычислений эти задачи не решить. Мы показали реальность таких вычислений на примере сеточной задачи, которая в принципе может быть распараллелена и по ячейкам реактора при наличии нескольких тысяч процессоров, и по узлам сетки, если их десятки тысяч. Сложные программы вычисления ячейечных характеристик в узлах сетки без труда можно разместить здесь же или в соседних процессорах. Пока неясно, насколько трудным окажется распараллеливание теплогидравлических программ, но нет оснований сомневаться в принципиальной разрешимости и этой задачи. Основные препятствия на пути создания моделирующего комплекса скорее всего будут иметь организационно-финансовый характер.

Автор искренне признателен коллективу Научного центра нейрокомпьютеров Российской академии наук и его директору А.И. Галушкину за благожелательное содействие в выполнении работы.

СПИСОК ЛИТЕРАТУРЫ

1. О причинах и обстоятельствах аварии на 4 блоке Чернобыльской АЭС 26 апреля 1986 г. Доклад Комиссии Госпроматомнадзора СССР, 1991.
2. Городков С.С. Погрешность некоторых сеточных алгоритмов в расчете пространственной кинетики энергетических реакторов. — Атомная энергия, 1993, т. 74, вып. 5, с. 414—444.
3. Городков С.С. Использование векторного процессора в реакторных расчетах. — Там же, 1986, т. 61, вып. 4, с. 284—285.