АЛГОРИТМ СБОРКИ МИНИМАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ ФРАГМЕНТОВ В СБОРОЧНОЙ ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ

С.П. Нечаев

Новосибирский государственный университет

1. Введение

Сборочная технология представляет собой новый подход к созданию параллельных программ. Она заключается в том, что параллельная программа собирается из заранее заготовленных атомарных фрагментов вычислений. Преимущества такого подхода в следующем: достаточно один раз создать атомарный фрагмент, а потом можно в зависимости от конфигурации мультикомпьютера и размера задачи собирать из таких фрагментов параллельную программу, которая оптимальным образом использует доступные ресурсы [1].

Основные принципы сборочной технологии заключаются в следующем:

• Программа конструируется из небольших атомарных фрагментов вычислений (код + обрабатываемые им данные), далее фрагментов.

• Фрагментированность программы сохраняется в ходе вычислений.

• Фрагменты программы в ходе вычислений мигрируют между процессорными элементами (ПЭ) мультикомпьютера для обеспечения равной загрузки всех его ресурсов.

В программировании численных моделей основным атомарным фрагментом является ячейка (узел) сетки (все переменные ячейки и код их обработки). Небольшой размер фрагмента имеет следствием простоту конструирования программы (ее сборки). Однако качество исполнения собираемой программы (ее производительность, потребляемые ресурсы и т.д.), собранной из атомарных фрагментов-ячеек, оказывается низким из-за больших затрат на реализацию управления и структур данных. Для повышения качества программ необходимо собирать их из фрагментов подходящих размеров и структуры [2].

2. Динамическая балансировка загрузки

Известно, что равная загрузка ПЭ мультикомпьютера способствует повышению эффективности параллельной программы. Некоторые вычисления, например метод частиц в ячейках или адаптивные разностные схе-

мы, устроены так, что загруженность ПЭ во время работы программы меняется непредсказуемым образом. Поэтому для эффективной работы программы требуется динамическая балансировка загрузки. При использовании сборочной технологии для обеспечения динамической балансировки загрузки нужно поддерживать фрагментированную структуру программ во время их выполнения. Но если поддерживать фрагментацию на уровне атомарных фрагментов, то затраты на это превысят получаемый от такого подхода выигрыш. Поэтому предлагается двухуровневая фрагментация программ.

Сначала пользователем создается атомарный фрагмент – фрагмент, который отражает смысл поставленной задачи. Размер атомарного фрагмента должен быть как можно меньше, чтобы упростить формулировку задачи в терминах программы. Для уменьшения потерь собираемой программы на реализацию управления из атомарных фрагментов собирается минимальный фрагмент. При этом его вид определяется особенностями задачи и архитектуры вычислительной системы. В численных алгоритмах это может быть, например, слой или колонка ячеек. Сборка минимального фрагмента выполняется до начала работы программы, а во время вычислений он неделим. Таким образом, собираемая программа описывается в терминах атомарных фрагментов, а на исполнение идет программа, описанная в терминах минимальных фрагментов, в которых уже частично учтены особенности и задачи, и вычислителя.

Во время работы программы полученные минимальные фрагменты могут перемещаться от перегруженного ПЭ к недогруженному для выравнивания загрузки [1].

3. Сборка

При сборке всем тем переменным атомарных фрагментов, которые соответствуют значению одной и той же функции в одной и той же точке пространства, ставится в соответствие одна переменная минимального фрагмента и указывается правило, по которому значение новой переменной вычисляется как функция от значений исходных (рис. 1).

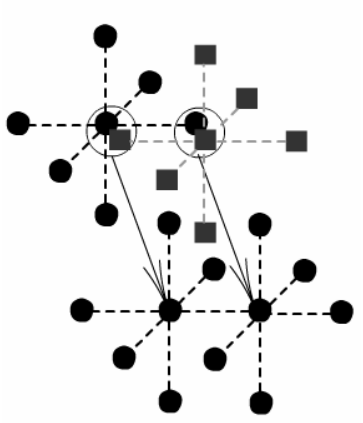

Рис. 1. Сборка

Рис 1. Сборка.

4. Ограничения

Созданный алгоритм работает при следующих достаточно серьезных ограничениях:

1. Вычисления производятся в прямоугольной системе координат.

2. Код атомарного фрагмента изменяет значение не более одной переменной.

Последнее ограничение связано с тем, что один и тот же элемент

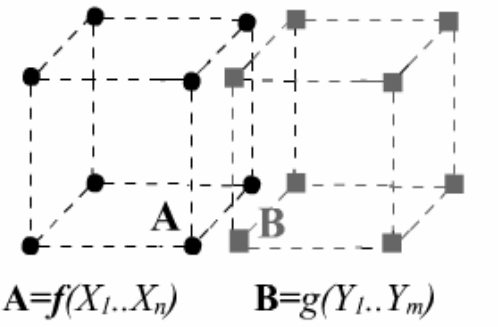

Рис. 2 ...менных

массива, полученный в результате сборки, должен обрабатываться двояким образом: как переменная A клетки одного фрагмента и как переменная B соседнего. В общем случае значения этих переменных вычисляются как разные функции от разных аргументов, поэтому нужно адекватным образом отразить вклад каждой из собираемых переменных в значение получающегося элемента массива.

5. Описание алгоритма

Основная задача заключалась в разработке алгоритма, который по заданному алгоритму сборки параллельной программы и коду атомарного фрагмента генерирует минимальный фрагмент, т.е. конструирует код и переменные.

Следующий пример иллюстрирует работу алгоритма. Пусть задан атомарный фрагмент – шаблон сетки, код, его обрабатывающий, размер и структура минимального фрагмента, который необходимо сконструировать. Предполагается, что минимальный фрагмент всегда имеет прямоугольную структуру (полоса или слой атомарных структур). Размер выражается числом атомарных фрагментов, собираемых по соответствующему направлению. Например, размер 4×2×3 означает, что минимальный фрагмент собирается из четырех атомарных фрагментов по оси X, двух по Y и трех по Z.

На рис. 3 изображены атомарный фрагмент и минимальный фрагмент, который требуется собрать.

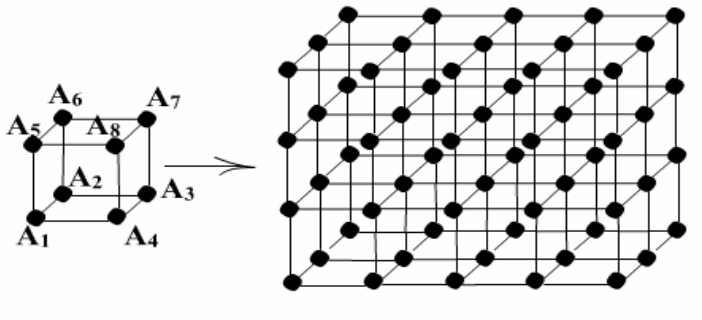

Рис. 3 Атомарный и минимальный фрагменты

Алгоритм состоит из следующих этапов:

На первом этапе для каждого из трех направлений (X, Y, Z) строятся списки переменных, которые должны стать элементами одного и того же массива. Вычисляются размеры каждого из массивов. Для иллюстрации этого этапа присоединим к исходному атомарному фрагменту по такому же фрагменту с каждого из трех направлений.

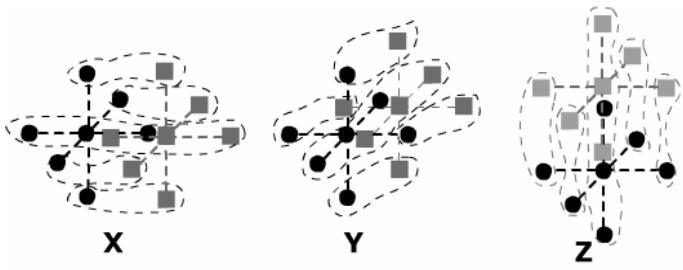

Рис. 4 Переменные – элементы одного массива

На рис. 4 обведены те переменные, которые должны быть элементами одного массива по соответствующему направлению.

На втором этапе выбирается переменная исходного атомарного фрагмента, которая «связывает» все три направления. Эта переменная должна входить во все три массива, причем в каждый из массивов еще как минимум с одной переменной из того же самого фрагмента. На рис. 5 эта пере-

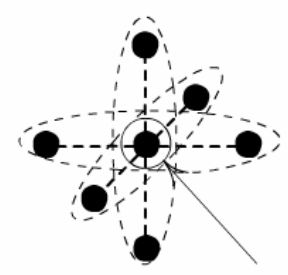

Рис. 5 Переменная, связывающая все три направления

менная обведена сплошной линией. Если такой переменной не найдется, значит, по данной схеме сборку невозможно произвести.

На третьем этапе объявляется массив (размеры его были вычислены ранее), добавляется заголовок цикла, который будет обрабатывать собранную область, и затем по всему коду исходного атомарного фрагмента имена переменных заменяются на соответствующие имена элементов массива. Предполагается, что наиболее часто в качестве минимальных фрагментов будут использоваться «полоса» и «слой» (рис. 6). Удобно будет их рассматривать как трехмерные массивы размеров $a_1 \times a_2 \times a_3$, где $a_i \in N$, $\exists!\ a_i \neq 1$ для полосы и вида $a_1 \times a_2 \times a_3$, где $a_i \in N$, $\exists!\ a_i \neq 1$ для слоя.

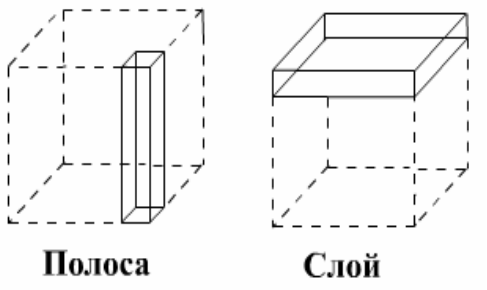

Рис. 6 Основные виды минимальных фрагментов

Заключение

Сборочная технология – перспективный подход к созданию высокоэффективных параллельных программ.

Преимущества сборочной технологии следующие:

1. Эффективное использование ресурсов. Фрагментированная структура программы обеспечивает возможность перемещения вычислений от перегруженных ПЭ к недогруженным для балансировки загрузки.

2. Автоматическая сборка программ. Пользователю необязательно обладать узкоспециальными знаниями по параллельным вычислениям. Он пишет лишь последовательный код своей программы или ее части, а параллельный код генерируется автоматически.

3. Простота отладки. Поскольку пользователь имеет дело только с кодом атомарного фрагмента, а его объем много меньше объема кода всей программы, то отладить и оптимизировать его значительно проще. Разработанный алгоритм реализует один из ключевых моментов сборочной технологии – сборку минимального фрагмента вычислений. Разработана программа, реализующая данный алгоритм. В дальнейшем она будет встроена в систему сборочного асинхронного параллельного программирования.

Литература

1. Kraeva M., Malyshkin V. Implementation of PIC Method on MIMD Multicomputers with Assembly Technology //Proc. of HPCN Europe 1997 (High Performance Computing and Networking) Int. Conference, LNCS. – Springer Verlag, 1997. – Vol.1255. – P. 541–549.
2. Kraeva M.A., Malyshkin V.E. Assembly Technology for Parallel Realization of Numerical Models on MIMD-Multicomputers //In the special issue of the International Journal on Future Generation Computer Systems, devoted to Parallel Computing Technologies. – 2001. – Vol. 17, № 6. – P. 755–765.