Runtime-система FP RTS управления исполнением фрагментированных программ на мультикомпьютере

Г.А. Щукин
Новосибирский государственный технический университет

Разработанная runtime-система FP RTS обеспечивает параллельное исполнение фрагментированных программ на мультикомпьютере. В статье показано устройство FP RTS и приведены тесты исполнения фрагментированных вычислительных алгоритмов.

Введение

Развитие и распространение высокопроизводительных параллельных вычислительных систем ставит задачу не только создания соответствующего эффективного параллельного программного обеспечения для решения вычислительных задач, но и оптимального использования вычислительного оборудования.

Одной из технологий, позволяющей не только упростить процесс создания параллельных программ и вывести его на новый уровень, но и обеспечить оптимальное использование вычислительных ресурсов, является технология фрагментированного программирования [1, 2].

Во фрагментированном программировании программа состоит из фрагментов данных (аналог переменных программы) и фрагментов кода (действия, выполняемые над данными); через применение определенных фрагментов кода к определенным фрагментам данных получаются фрагменты вычислений, а исполнение программы заключается в запуске всех ее фрагментов вычислений.

FP RTS

Наличие runtime-системы [3] для исполнения фрагментированных программ позволяет автоматически использовать динамические свойства фрагментированной программы: подстройка под ресурсы вычислительной системы для оптимального исполнения через изменение размера и числа фрагментов данных; динамическая балансировка нагрузки через перераспределение фрагментов вычислений между вычислительными узлами; коммуникации (обмен данными) на фоне вычислений. Таким образом, программист выполняет только описание фрагментированной программы, тогда как исполнение программы вынесено отдельно в runtime-систему.

Разработанная автором runtime-система FP RTS осуществляет динамическое управление исполнением фрагментированных программ на вычислительных машинах широкого класса – мультипроцессорах (SMP) и мультикомпьютерах (MPP, кластеры). Для функционирования runtime-системы на каждом вычислительном узле запускается менеджер FP RTS (рис. 1), который можно разделить на 3 подсистемы.

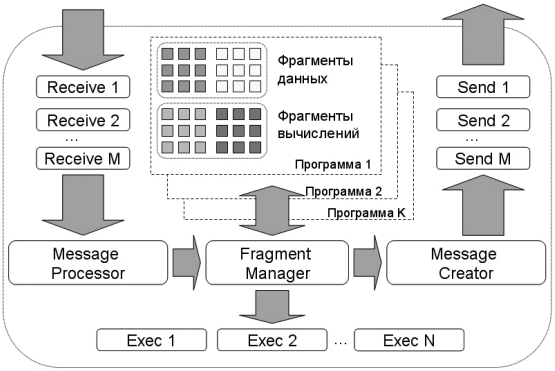


Рис. 1. Структура FP RTS на одном вычислительном узле

Подсистема хранения содержит исполняющиеся в текущий момент фрагментированные программы.

В начале работы каждой программы FP RTS осуществляет автоматическое распределение фрагментов данных и вычислений по узлам. Таким образом, на каждом узле хранится только часть фрагментов данных и вычислений из всей программы; некоторые фрагменты данных могут дублироваться на разных узлах. Во время исполнения программы фрагменты данных и вычислений могут перемещаться между узлами.

Подсистема исполнения содержит пул исполнительных потоков для запуска фрагментов вычислений на узле.

Каждый фрагмент вычислений имеет входные и выходные фрагменты данных. Минимальное условие для запуска фрагмента вычислений – готовность всех его входных фрагментов данных. После исполнения фрагмента вычислений вычисленные им выходные фрагменты данных могут быть переданы на вход другим фрагментам вычислений. Кроме зависимостей по данным, порядок исполнения фрагментов вычислений в общем случае не детерминирован, но может регулироваться с помощью системы динамических приоритетов.

На каждом узле готовые к исполнению фрагменты собираются в группы (группировка по фрагментам кода) и для каждой группы вы-

числяется динамический приоритет на основе приоритетов содержащихся в ней фрагментов (приоритет фрагмента может зависеть от его «важности», вычислительного веса и т.п.). Приоритет также вычисляется и для всей программы. В итоге на исполнение берутся фрагменты с максимальным приоритетом из программы и группы с максимальными приоритетами.

Подсистемы хранения и исполнения в одиночку могут обеспечить функционирование FP RTS и запуск фрагментированных программ на машинах с общей памятью (SMP).

Сетевая подсистема отвечает за взаимодействие между менеджерами FP RTS на разных узлах, а также за пересылку фрагментов данных и вычислений между узлами. Вся информация передается в виде сообщений. На каждом узле запускаются отдельные потоки для отправки и приема сообщений. Потоков отправки/приема может быть несколько, что обеспечивает параллельные отправку/прием нескольких сообщений.

Для описания и запуска фрагментированных программ в FP RTS разработан специальный интерфейс (API) – набор классов и функций, использующихся для описания фрагментов вычислений и фрагментированных программ, запуска/остановки runtime-системы и загрузки в нее программ для исполнения. Интерфейс может использоваться как программистами фрагментированных программ, так и компилятором/транслятором с языка фрагментированного программирования высокого уровня.

Результаты экспериментов

Тестирование проводилось на кластере НКС-30Т; на каждом вычислительном узле кластера находятся два 4 ядерных процессора Intel Xeon E5450 3 GHz с общей памятью 16 Gb. На каждом вычислительном узле запускалось по одному менеджеру FP RTS (процессу).

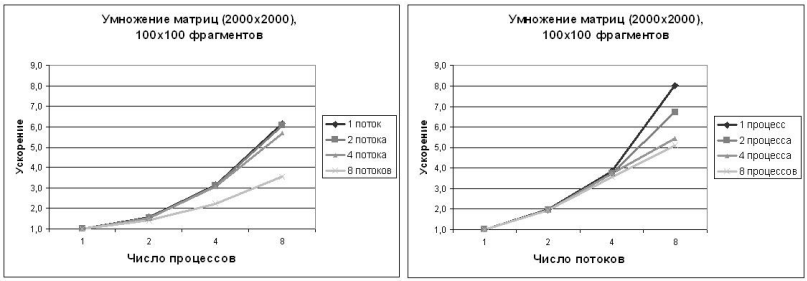


Рис. 2. Умножение матриц: ускорение

Фрагментированное умножение матриц представлено на рис. 2.

Ускорение достигается как при увеличении числа вычислительных узлов, так и при увеличении числа исполнительных потоков. Ускорение тем лучше, чем больше доля вычислений и, соответственно, меньше доля накладных расходов на управление.

Фрагментированное LU-разложение представлено на рис. 3-4.

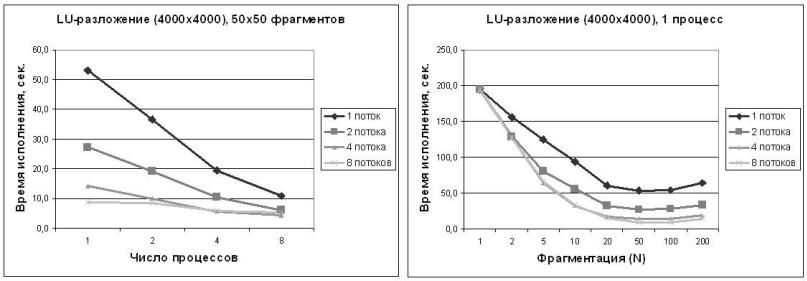


Рис. 3. LU-разложение: время исполнения и фрагментация

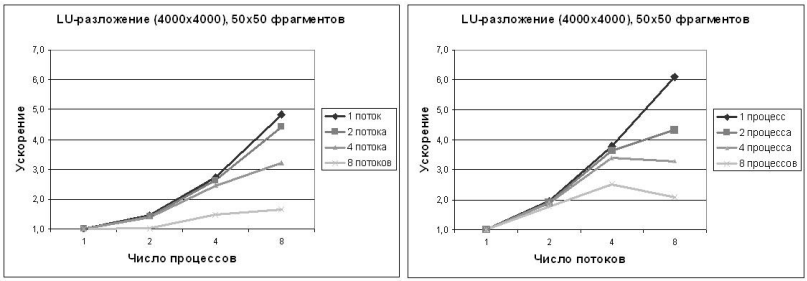


Рис. 4. LU-разложение: ускорение

За счет фрагментации время исполнения программы существенно снижается. Несмотря на предусматриваемый фрагментированным алгоритмом большой объем передаваемых между узлами фрагментов данных, достигается неплохое ускорение при увеличении числа процессов и потоков. Некоторое падение ускорения при большом числе процессов и потоков можно объяснить снижением относительной доли вычислений для параллельного исполнения всеми потоками, а также техническими ограничениями вычислительного узла (невозможностью предоставить ресурсы всем потокам).

Литература

1. **Вальковский В.А., Малышкин В.Э.** Синтез параллельных программ и систем на вычислительных моделях. Новосибирск: Наука, 1988. С. 128.
2. **Malyshkin V.E.** Fragmented Programming of Library Parallel Numerical Subroutines // Proceedings of the 7-th Int. conference on New Trends in Software Methodologies, Tools and Techniques, IOS Press, 28-30 September, 2007. Dubai, 2007. Vol. 193. P. 425-430.
3. **Kalgin K.V., Malyshkin V.E., Nechaev S.P., Tschukin G.A.** Runtime System for Parallel Execution of Fragmented Subroutines// PACT-2007, LNCS. 2007. Vol. 4671. P. 544-552.