ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

УДК 004.272.45

С. В. Волобуев, И. В. Зотов, В. Н. Николаев

ПРОЦЕДУРА РАСПРЕДЕЛЕННОЙ ПАРАЛЛЕЛЬНО-КОНВЕЙЕРНОЙ БАРЬЕРНОЙ СИНХРОНИЗАЦИИ, ИНВАРИАНТНАЯ К СПОСОБУ РАЗМЕЩЕНИЯ СИНХРОНИЗИРУЕМЫХ ПРОЦЕССОВ

Предложена процедура распределенной параллельно-конвейерной барьерной синхронизации для однородных многопроцессорных систем, инвариантная к способу размещения синхронизируемых процессов в системе и применимая к классу матричных топологических структур. Приведены результаты имитационного моделирования процедуры.

Ключевые слова: *многопроцессорная система*, *матричная топология*, *межпроцессорное взаимодействие*, *координация параллельных процессов*, *барьерная синхронизация*, *имитационное моделирование*, *Q-схема.*

Введение. Одним из факторов, сдерживающих рост производительности многопроцессорных вычислительных систем (МВС) широкого класса, является необходимость регулярной межпроцессорной барьерной синхронизации [1]. С целью снижения временных затрат на синхронизацию при выполнении задач в ряде современных МВС используются аппаратные решения, многие из которых имеют распределенный характер и позволяют реализовать оперативную параллельную синхронизацию для нескольких барьеров [2—6]. Однако известные варианты решения характеризуются невысокой гибкостью, которая, с одной стороны, проявляется в наличии ограничений на способ размещения в МВС синхронизируемых процессов [2, 4], а с другой — обусловлена привязкой к конкретной топологической структуре системы [3, 5, 6].

В [7] авторами была разработана процедура параллельно-конвейерной барьерной синхронизации с использованием распределенной координирующей среды, не накладывающая ограничений на способ размещения синхронизируемых процессов и ориентированная на МВС с двумерной матричной топологией. В настоящей работе описан расширенный вариант этой процедуры, применимый к более общему классу топологий МВС. Кроме того, анализируются результаты вычислительного эксперимента, проведенного с целью исследования ряда динамических характеристик разработанной процедуры синхронизации, на примере МВС с двумерной матричной топологией.

Топологическая модель МВС. Представим МВС в виде графа $H =< M, U >$, множество вершин M которого соответствует множеству однотипных процессорных элементов (ПЭ) — модулей системы, а множество дуг $U \subseteq M \times M$ отражает связи между ПЭ. Зададим для каждого ПЭ составной порядковый номер $(x_1, x_2, .., x_d)$, $x_i = \overline{1, N_i}$, $i = \overline{1, d}$, где d — размер-

ность МВС (для топологии кольца $d=1$, для двумерной матричной топологии — $d=2$ и т.д.). Вершины графа H сопоставим множеству ПЭ системы так, что

$$\left.\begin{array}{l}\left(m\left(x_1,x_2,..,x_d\right),m\left(x_1+1,x_2,..,x_d\right)\right)\in U,\ \ x_1=\overline{1,(N_1-1)},\ \ x_i=\overline{1,N_i},\ \ i=\overline{1,d};\\ \left(m\left(x_1,x_2,..,x_d\right),m\left(x_1,x_2+1,..,x_d\right)\right)\in U,\ \ x_2=\overline{1,(N_2-1)},\ \ x_i=\overline{1,N_i},\ \ i=(1,3,..,d);\\ \ldots\\ \left(m\left(x_1,x_2,..,x_d\right),m\left(x_1,x_2,..,x_d+1\right)\right)\in U,\ \ x_d=\overline{1,(N_d-1)},\ \ x_i=\overline{1,N_i},\ \ i=\overline{1,(d-1)},\end{array}\right\}$$

где $m(x_1,x_2,...,x_d)$ — ПЭ с порядковым номером $(x_1,x_2,...,x_d)$. Систему, представленную описанным способом, обозначим как $N^{(H)}$.

При таком описании топологические структуры МВС с обратными связями (кольцо, тор, трехмерный тор и т.д.) и без них (разорванное кольцо, матрица, куб) идентичны и трансформируются в одну систему $N^{(H)}$. В отличие от систем без обратных связей, в топологических структурах с обратными связями начальный порядковый номер может быть присвоен любому ПЭ.

Процедура синхронизации. Для обеспечения барьерной синхронизации в МВС вводится обособленная однородная координирующая среда. Каждому ПЭ ставится в соответствие ячейка координирующей среды, соединенная, согласно топологии МВС, с соседними ячейками, что позволяет сохранить однородность системы и обеспечить ее масштабируемость. В процессе функционирования МВС каждому барьеру назначается одноразрядный слой координирующей среды для распространения признаков достижения барьера.

Определение 1. Будем считать, что ПЭ $m'(x'_1,x'_2,..,x'_d)$ имеет меньший порядковый номер, чем ПЭ $m''(x''_1,x''_2,..,x''_d)$, если и только если

$$(\forall x'_i,x''_i,i=\overline{1,d}:x'_i\le x''_i)\wedge(\exists x'_i,x''_i,i=\overline{1,d}:x'_i\ne x''_i).$$

Определение 2. Соответственно будем считать, что ПЭ $m'(x'_1,x'_2,..,x'_d)$ имеет больший порядковый номер, чем ПЭ $m''(x''_1,x''_2,..,x''_d)$, если и только если

$$(\forall x'_i,x''_i,i=\overline{1,d}:x'_i\ge x''_i)\wedge(\exists x'_i,x''_i,i=\overline{1,d}:x'_i\ne x''_i).$$

На основании этих определений можно сформулировать правило формирования признака достижения барьера для каждой ячейки среды следующим образом: если соседние ПЭ с меньшим порядковым номером завершили свои параллельные ветви программы (или не участвуют в синхронизации для соответствующего барьера), и при этом текущий ПЭ также завершил свою ветвь (или не участвует в синхронизации), то на выходе ячейки устанавливается признак достижения барьера, затем с выхода текущей ячейки он передается ячейкам, подключенным к ПЭ с большими порядковыми номерами. Формирование признака достижения барьера на выходе ячейки с максимальным порядковым номером будет означать, что все процессоры достигли барьера. Далее признак достижения барьера распространяется в обратном направлении и запускает ожидающие процессоры.

Для уменьшения числа линий связи между ячейками слои в координирующей среде разбиты на группы. В общем случае n слоев разделены на p групп по m слоев в каждой. В один и тот же момент времени в i-й группе происходит формирование признака достижения барьера, а в $(i+1)$-й — распространение признака. Остальные группы в это время не активны. Переключение групп слоев происходит последовательно циклически через определенные интервалы времени.

Экспериментальное исследование процедуры синхронизации. Для исследования динамических характеристик разработанной процедуры, в частности среднего времени синхронизации $T_{\text{ср}}$, было проведено имитационное моделирование распределенной координирующей среды. При этом в качестве модели МВС была взята двумерная матричная топология. В ходе моделирования варьировались способы размещения множеств синхронизируемых $J(v_s)$ и ожидающих $F(v_s)$ ветвей (v_s — один из барьеров) и определялись случайные значения времени завершения ветвей $B_i \in J(v_s)$. Здесь и далее используется терминология и математический аппарат, введенные в работе [7].

Вычислительный эксперимент был поставлен в графической системе моделирования Visual QChart Simulator [8]. Координирующая среда была представлена как сеть массового обслуживания, в которой объектами обслуживания являются признаки достижения барьера и иные координирующие сигналы.

Значение $T_{\text{ср}}$ определялось по формуле

$$T_{\text{ср}} = t^+ + \Delta t + t^-,$$

где t^+ — интервал времени от момента завершения последней ветви из множества $J(v_r)$ до начала распространения сигнала достижения барьера через координирующую среду, Δt — время распространения сигнала достижения барьера, t^- — интервал времени от момента получения процессором, выполняющим последнюю ветвь из множества $F(v_r)$, сигнала достижения барьера до момента активизации данной ветви. Поскольку величина интервала $t^+ + t^-$ не зависит от способа размещения ветвей на множестве модулей МВС и при $n \le 256$ не превышает 16 задержек вентиля, в ходе моделирования оценивалась только величина Δt.

Условия проведения эксперимента были определены следующим образом. Количество множеств $J(v_s)$ и $F(v_s)$, а также их мощность устанавливались постоянными на время одного этапа моделирования.

Время выполнения ветвей $B_i \in J(v_s) \cup F(v_s)$ задавалось случайно в диапазоне (1—12)$T_{\max}$ ($T_{\max}$ — максимальное время синхронизации). В случае, когда время моделирования превышало 5 часов, максимальное время выполнения ветвей снижалось до $6T_{\max}$. Среднее значение Δt определялось как среднее арифметическое не менее чем по 30 итерациям.

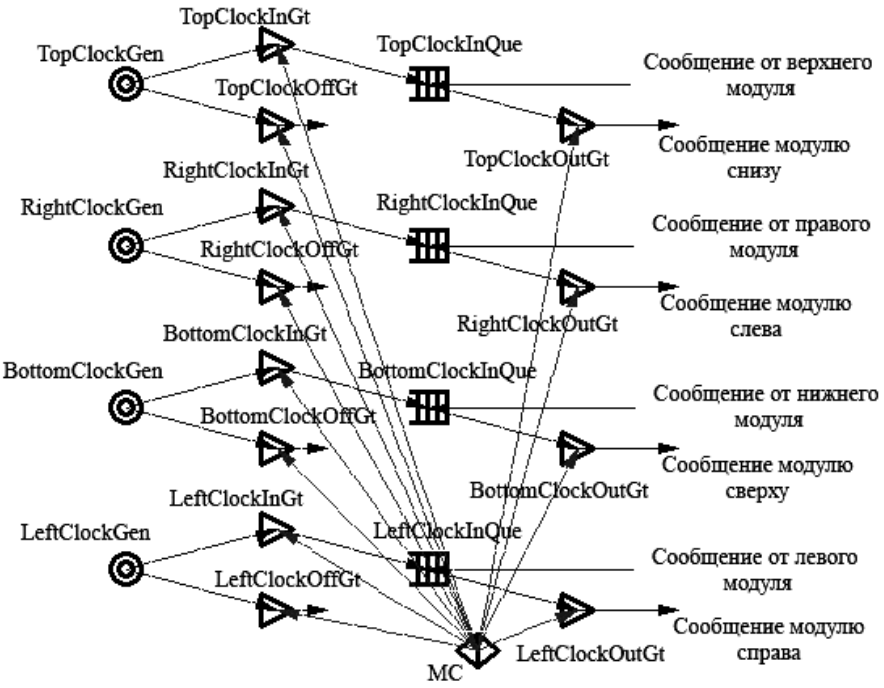
В ходе моделирования вся информация записывалась в текстовый файл. По окончании моделирования данный файл содержал следующие результаты: количество достигнутых барьеров, время синхронизации каждого барьера, среднее время синхронизации.

Модель координирующей среды. В ходе подготовки вычислительного эксперимента были построены Q-схемы, моделирующие работу сред размера $N = 6,8,12,14,16,24,28$. Q-схема одной ячейки среды представлена на рисунке.

Рассматриваемая схема состоит из следующих элементов.

Генераторы TopClockGen(i), RightClockGen(i), BottomClockGen(i), LeftClockGen(i) предназначены для формирования заявок, имитирующих сигналы синхронизации, которые распространяются от модулей, расположенных сверху, справа, снизу, слева от текущего модуля соответственно. Время генерации заявок постоянно и составляет один такт моделирования. Очереди TopClockInQue(i), RightClockInQue(i), BottomClockInQue(i), LeftClocknQue(i) служат для хранения заявок, пришедших от модулей, расположенных сверху, справа, снизу, слева от текущего соответственно. Клапаны TopClockInGt(i), RightClockInGt(i), BottomClockInGt(i), LeftClockInGt(i) предназначены для разрешения/запрета прохождения заявок между соответствующими генератором и очередью. Клапаны TopClockOffGt(i), RightClockOffGt(i), Bottom-

ClockOffGt(i), LeftClockOffGt(i) предназначены для удаления из системы заявок, поступающих от соответствующих генераторов.



Совокупность генератора *Gen(i), клапанов *InGt(i) и *OffGt(i) можно представить управляемым устройством, моделирующим поступление фиктивных сигналов синхронизации к модулям, расположенным по границам матричной структуры. Для модулей, расположенных в верхней строке, имитируется подача сигнала синхронизации сверху, в нижней строке — снизу, в правом и левом столбцах — соответственно справа и слева. Клапаны TopClockOutGt(i), RightClockOutGt(i), BottomClockOutGt(i), LeftClockOutGt(i) предназначены для разрешения/запрета распространения заявок от текущего модуля к модулям, расположенным ниже, левее, выше и правее соответственно. Пара клапанов TopClockOutGt(i), RightClockOutGt(i) открывается одновременно при условии, что очереди TopClockInQue(i), RightClockInQue(i) заняты. Аналогично осуществляется взаимодействие клапанов BottomClockOutGt(i), LeftClockOutGt(i) и очередей BottomClockInQue(i), LeftClocknQue(i). Контроллер MC(i) управляет открытием клапанов i-го модуля Q-схемы в зависимости от расположения модуля в системе и моментов прихода сигналов синхронизации от соседних модулей.

Результаты вычислительного эксперимента. Моделирование зависимости времени синхронизации Δt от количества одновременно достигнутых барьеров n проводилось при $p = 2$ и $|J(v_s)| = |F(v_s)| = 5$. Результаты подтвердили отсутствие зависимости Δt от n, поэтому исследование остальных характеристик проводилось при фиксированном значении n.

Моделирование зависимости времени Δt от мощности множеств $J(v_s)$ и $F(v_s)$ (от размерности барьера) проводилось при $n = p = 2$. В результате разница между максимальным и минимальным временем синхронизации для системы с $N = 16$ составила $\approx 22\,\%$, а для $N = 24$ — $\approx 34\,\%$. Данные отклонения обусловлены зависимостью Δt от расположения последних ветвей множеств $J(v_s)$ и $F(v_s)$ в модулях МВС. Параметр Δt можно представить следующим образом:

$$\Delta t = \Delta t' + \Delta t'' + \Delta \bar{t},$$

где $\Delta t'$ — время распространения признака достижения барьера от модуля МВС, выполнившего последнюю ветвь множества $J(v_s)$, до (N, N)-го модуля; $\Delta t''$ — время распространения

сигнала запуска от (N,N)-го модуля МВС, до модуля, выполняющего последнюю ветвь множества $F(v_s)$; $\Delta\bar{t}$ — время до активизации слоя, отвечающего за синхронизацию барьера v_s. При этом значение $\Delta\bar{t}$ случайно и в среднем равно

$$\Delta\bar{t} = \frac{pt_a}{2},$$

где t_a — интервал времени активности одного слоя.

Зависимость Δt от расположения ветвей множества $J(v_s) \cup F(v_s)$ среди модулей МВС снижается при увеличении p, поскольку при $p \to \infty$ $\dfrac{\Delta t' + \Delta t''}{\Delta\bar{t}} \to 0$. Значения интервалов времени $\Delta t'$ и $\Delta t''$ зависят от расположения ветвей множества $J(v_s) \cup F(v_s)$. Так, при $|J(v_s)| = |F(v_s)| = 1$ среднее значение $\Delta t'$ и $\Delta t''$ равно Nt_m, где t_m — время распространения сигнала синхронизации через одну ячейку координирующей среды. Если $|J(v_s)| \to N^2$ и $|F(v_s)| \to N^2$, то $\Delta t', \Delta t'' \to (2N-1)t_m$.

Исследование среднего времени Δt от размера МВС N проводилось при значениях $p = 4$, $n = 8$. Для исключения влияния мощности множеств $J(v_s)$ и $F(v_s)$ на результаты измерений было принято $|J(v_s)| = |F(v_s)| = 0{,}9\,N^2/n$. Результаты моделирования подтвердили линейное увеличение времени Δt от размера системы. Увеличение N в 2 раза приводит практически к двукратному увеличению Δt. Так, при $N = 14$ $\Delta t \approx 1{,}99$ мкс, а при $N = 28$ — $\Delta t \approx 3{,}7$ мкс (при задержке двухходового логического элемента, равной 5 нс). Полученные значения соизмеримы с известными распределенными аппаратными решениями [2, 4, 6].

Выводы. Как показали результаты вычислительного эксперимента, разработанная процедура характеризуется отсутствием зависимости времени синхронизации от количества одновременно достигаемых барьеров, линейным ростом времени синхронизации от размера МВС, а также его зависимостью от числа участвующих в барьере процессоров (снижающейся при увеличении количества групп барьеров). Процедура применима к матричным топологиям различной размерности как с незамкнутыми, так и с замкнутыми границами и инвариантна к способу размещения синхронизируемых и ожидающих ветвей в системе.

Работа выполнена при поддержке гранта Президента РФ МД-685.2009.8.

СПИСОК ЛИТЕРАТУРЫ

1. *O'Boyle M.*, *Stohr E.* Compile time barrier synchronization minimization // IEEE Transactions on Parallel and Distributed Systems. 2002. Vol. 13, N 6. P. 529—543.

2. *Delgado M.*, *Kofuji S.* A distributed barrier synchronization solution in hardware for 2D-mesh multicomputers // Proc. 3rd Intern. Conf. High Performance Computing. 1996. P. 368—373.

3. *Yang J.-S.*, *King C.-T.* Designing Tree-Based Barrier Synchronization on 2D Mesh Networks // IEEE Transactions on Parallel and Distributed Systems. 1998. Vol. 9, N 6. P. 526—534.

4. *Ramakrishnan V.*, *Scherson I. D.*, *Subramanian R.* Efficient techniques for nested and disjoint barrier synchronization // J. of Parallel and Distributed Computing. 1999. Vol. 58, N 8. P. 333—356.

5. *Cohen W. E.*, *Hyde D. W.*, *Gaede R. K.* An optical bus-based distributed dynamic barrier mechanism // IEEE Transactions on Computers. 2000. Vol. 49, N 12. P. 1354—1365.

6. *Moh S.*, *Yu C.*, *Lee B.* et al. Four-ary tree-based barrier synchronization for 2D meshes without nonmember involvement // IEEE Transactions on Computers. 2001. Vol. 50, N 8. P. 811—823.

7. *Волобуев С. В.*, *Зотов И. В.* Организация параллельно-конвейерной барьерной синхронизации в матричных многопроцессорных системах на основе распределенной координирующей среды // Параллельные вычисления и задачи управления (PACO'08). М.: Институт проблем управления им. В.А. Трапезникова РАН, 2008. С. 616—642.

8. *Зотов И. В.* и др. Визуальная среда имитационного моделирования VisualQChart. Свид. об официальной регистрации программы для ЭВМ №2007611310 от 27.03.07.

Сведения об авторах

Сергей Викторович Волобуев — аспирант; Курский государственный технический университет, кафедра вычислительной техники; E-mail: magehunter@rambler.ru

Игорь Валерьевич Зотов — д-р техн. наук, профессор; Курский государственный технический университет, кафедра вычислительной техники; E-mail: zotovigor@yandex.ru

Виктор Николаевич Николаев — д-р техн. наук, профессор; Курский государственный технический университет, кафедра информационных систем и технологий; E-mail: nikovic@yandex.ru

Рекомендована кафедрой вычислительной техники

Поступила в редакцию 29.12.09 г.