УДК 681.3.06

СТИРЕНКО С.Г.,
МИТИН С.В.

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ ПРИ РАСПОЗНОВАНИИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ КЛЕТОЧНОЙ НЕЙРОННОЙ СЕТИ

Предложен подход для эффективной реализации параллельных вычислений на основе клеточной нейронной сети. Исследование выполнялось с использованием изображений в оттенках серого цвета. Описанный подход подразумевает использование SPMD парадигмы для распараллеливания данных. Процесс распараллеливания осуществлялся при помощи интерфейса передачи сообщений MPI, а производительность оценивалась реализацией на двух кластерных архитектурах.

In this paper a simple but effective approach for parallelization of cellular neural networks for image processing is developed. Digital gray-scale images were used to evaluate the program. The approach uses the SPMD model and is based on the structural data parallel approach. The process of parallelizing the algorithm employs HPF to generate an MPI-based program and the performance behavior was analyzed on two different cluster architectures.

Введение

Клеточная нейронная сеть (*Cellular Neural Network*) (КНС), объединяет в себе черты клеточных автоматов (локальность взаимодействия) и нейронных сетей и является элементарной математической моделью, ячейкой которой является аппаратная или программная модель нейрона. КНС может быть использована для широкого спектра задач, от приложений для сигнальных процессоров (выделение признаков, выявление движения и т.п.) и анализа трехмерный поверхностей (сцен) до решения дифференциальных уравнений в частных производных [1]. За последние годы в этой области появилось достаточно много публикаций [2-4].

В общем КНС может быть представлена как массив идентичных динамичных систем, которые локализуются подключением ячеек. Архитектурно КНС представляет собой объединение двух подходов: массивно-параллельных вычислений, типичных для полносвязных нейронных сетей и локальным взаимодействием между ячейками, характеризующий клеточные автоматы. Любая ячейка имеет связи только с соседними, взаимодействуя непосредственно только с ними. Ячейки не имеющие непосредственного контакта (находящиеся в окрестности) имеют косвенную связь.

Основной идеей представленной параллельной КНС является моделирование подхода решения задач с дискретным временем, которая может быть описана следующими выражениями: [5]

$$x_i(n+1) = \sum_{k \in N_r(j)} A_{jk} y_k(n) + \sum_{k \in N_r(j)} B_{jk} u_k(n) + I_j \quad (1)$$

$$y_i(n) = f[x_j(n)] \quad (2)$$

где x, y, u, I – состояния ячеек: выходные, входные, и значения по диагонали соответственно;

A и B – шаблон обратной связи окрестности и входной шаблон. Выходная функция определяется как $f(x)$, которая может быть функцией знака:

$$f(x) = \begin{cases} 1 \; при \; x \geq 0 \\ 0 \; при \; x < 0 \end{cases} \quad (3)$$

или кусочно-линейная функция выхода:

$$y_i(n) = 0.5*(||x(n)+1|| - ||x(n)-1|| \quad (4)$$

Этот подход довольно удобен для распараллеливания на классических супер-ЭВМ [6]. До сих пор не было сделано какого-либо серьезного анализа для парадигмы SPMD [7] (Single Program Multiple Data) КНС реализаций на кластерных архитектурах, а между тем, она значительно упрощает решение задач обработки изображений.

Механизм распараллеливания

При моделировании КНС будем использовать параллелизм на уровне данных. Этот подход упрощает процесс параллельного нейро-сетевого моделирования и приближает его по сложности к последовательному. Таким образом конечный пользователь имеет возможность более эффективно использовать высокопроизводительные параллельные вычисления работая с нейронными сетями.

Подход предполагает использование парадигмы SPMD, которая сочетает в себе возможности специализированных высокопроизводительных языков для распараллеливания процесса решения задачи. Процесс моделирования системы сводится к разработке последовательной программы, которая манипулирует распределенными данными. Поэтому трудоемкий процесс ручного физического распараллеливания смещается в сторону среды программирования и к компилятору. Этот подход с тем же успехом может быть применен при моделирования на супер-ЭВМ типа MPP и кластерных архитектурах.

Структурный подход параллелизма на уровне данных сводится к четырем стадиям, которые могут использоваться при переходе от последовательного описания программы к параллельной с целью эффективной реализации нейронной сети:

- Последовательное кодирование нейронной сети;
- Определение структуры данных;

- Выделение параллельных участков;
- Автоматизированная генерация параллельного кода и анализ.

С нашей точки зрения КНС можно представить в виде N-мерного регулярного (распознаваемого конечным автоматом) массива элементов. Определены три коррелированных массива: входной *u* (формула 1) и два выходных, один (*x*) для состояния ячеек в автоматное время *n* и второй (*y*) для состояния ячеек в предыдущий момент времени *n*–1. Каждый элемент массива *u* представляется одним пикселем входного изображения.

Придерживаясь SPMD модели сконцентрируем свое внимание на распараллеливании этих трех массивов. Предложим возможные схемы (рис. 1).

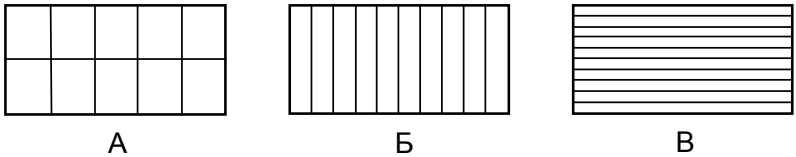


Рис. 1. Возможные схемы распараллеливания при обработки изображений для 10 процессоров: неперекрывающее расположение (А – поблочно, Б – поколоночно, В – построчно)

Исследование эффективности метода

Исследование включало в себя две стадии. На первой оценивались схемы распараллеливания данных изображения (см. рис. 1) и на второй стадии было проанализировано ускорение и масштабируемость лучшей схемы распараллеливания.

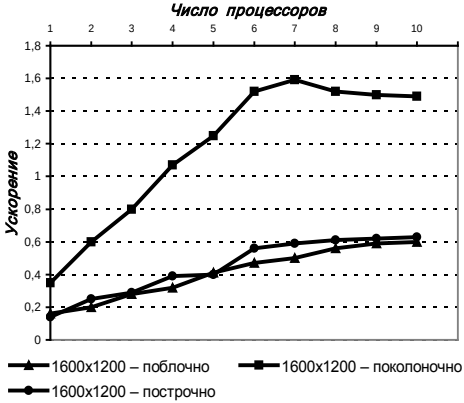
В ходе работы будем опираться на два различных критерия при анализе ускорения. С одной стороны – *абсолютное ускорение,* которое вычисляется сравнением времени выполнения параллельного программного кода на нескольких процессорах с временем выполнения последовательной программы на одном процессоре [8]. Таким образом, получим абсолютное ускорение за счет распараллеливания. С другой стороны – *относительное ускорение,* вычисляя время выполнения параллельной программы, запуская ее на нескольких процессорах с временем выполнения параллельной программы на одном процессоре.

Основой для работы являлся *исследовательский* кластер установленный в национальном техническом университете Украины «КПИ» [9]. Он имеет следующие характеристики: содержит восемьдесят четыре узла, каждый из которых имеет два двухядерных процессора Intel Xeon 5160 с тактовой частотой 3.00 ГГц и 4 ГБ DDR оперативной памяти, частота системной шины – 1333 МГц, коммутационная сеть – Infiniband 4x SDR.

Второй – *лабораторный* кластер был создан на базе научно-исследовательской лаборатории кафедры вычислительной техники НТУУ «КПИ»

[10] и имеет следующую конфигурацию: процессоры – Celeron 2.6 ГГц с 1 ГБ DDR оперативной памяти с частотой системной шины – 800 МГц, коммутационная сеть – Gigabit Ethernet.

Алгоритм КНС был запрограммирован с использованием интерфейса передачи сообщений MPI (реализация MPICH). Были допущены перекрытия размеров блоков на один пиксель с использованием параметра трансляции. Вдоль массива блочных границ, компилятор локализует вычислимую часть, которая, в противном случае, требовала бы передачу данных через коммутационную среду.



Оценка распараллеливания данных (стадия 1)

Оценка различных схем распараллеливания исследовался только на исследовательском кластере. В качестве входных данных использовалось стандартное изображение разрешением 1600×1200 пикселей. На рис. 2. показано абсолютное ускорение для различных схем распараллеливания. Легко заметить, что поколоночная схема распараллеливания имеет наилучшие результаты. С одной стороны это объясняется тем, что реже приходится обращаться к основной памяти, за счет, так называемого, «теплого кэша». Другими словами, более естественного механизма кэширования, т.е. потеря данных в кэш памяти минимальна. С другой стороны – поколоночное распараллеливание данных в соответствии с вычислительным потоком программы, обрабатывающей изображение, является более естественным.

Помимо этого можно увидеть, что издержки, связанные с дополнительной нагрузкой на коммуникационную среду и всю параллельную инфраструктуру в параллельной программе по сравнению с последовательной находится в интервале от 70 до 92 процентов. А также, что в худшем случае, разница между схема распараллеливания превышает 20 процентов. В результате анализа результатов показанных на рис. 2 можно сделать вывод, что распараллеливание нужно увязывать с соответствующей схемой распределения данных, а также с количеством узлов (процессоров) обработки.

Вращение изображения не дает эффекта ускорения обработки, если данные имеют поколоночное распределение. Имеется в виду, что разница во времени обработки различных частей изображения не влияет на время выполнения всей программы, при условии что данные распределены в

соответствии с поколоночной схемой. Хотя две другие схемы распределения данных имеют определенную зависимость времени выполнения программы от подстройки изображения.

Оценка ускорения и масштабируемости (стадия 2)

На второй стадии оценивалось ускорение и масштабируемость для изображений с различной разрешением. В соответствии с результатами первой стадии, использовалась схема только с поколоночным распределением данных.

Для исследования во второй стадии использовались оба кластера. Оценивалась параллельная обработка с использованием КНС для изображений от 400×300 до 4800×3600.

На рис. 3 показана абсолютное ускорение для изображений с разным разрешением. Лабораторный кластер показал ускорение более чем в два раза при использовании более чем трех вычислительных узлов (процессоров) для изображений более чем 800×600 пикселей.

На рис. 4 показано относительное ускорение параллельной программы для изображений с различным разрешением. Изображения с большим разрешением имеют лучшие результаты .

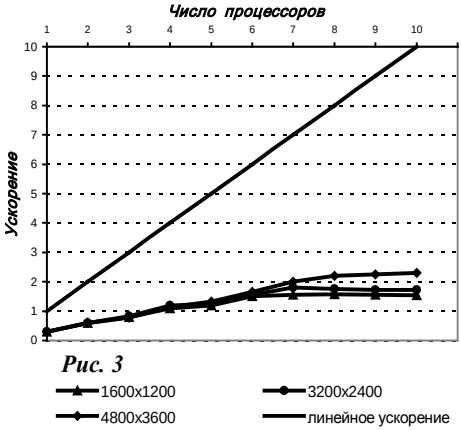


Рис. 3

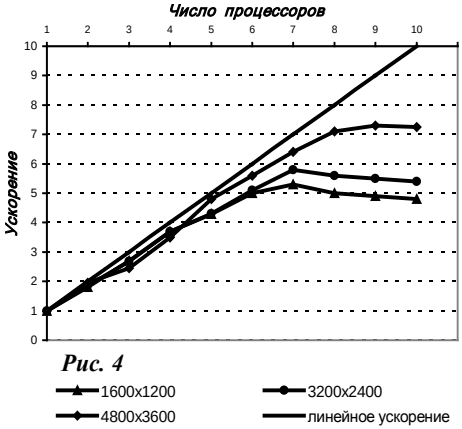


Рис. 4

Для больших изображений достигается более высокое значение ускорения. Однако, наблюдается некий процесс *насыщения*[1] абсолютного ускорения. Для изображений с меньшим разрешением насыщенность достигается раньше и дальнейшее увеличение числа узлов обработки не

[1] Под понятием насыщение понимается процесс прекращения роста параметра ускорения, при дальнейшем повышении количества узлов обработки.

приводит к повышению ускорения, поскольку коммуникационные затраты времени превышают время обработки. Таким образом, параметр ускорение существенно зависит от аппаратной и программной реализации коммуникационной среды (до 70%).

Программа показала довольно хорошую масштабируемость. При обработке КНС легко распараллеливается, тем самым, достигается довольно хорошая балансировка нагрузки между узлами.

Все, до сих пор, представленные результаты получены на лабораторном кластере [10]. Для верификации этих результатов, все оценки выполнялись повторно на более производительном исследовательском кластере [9].

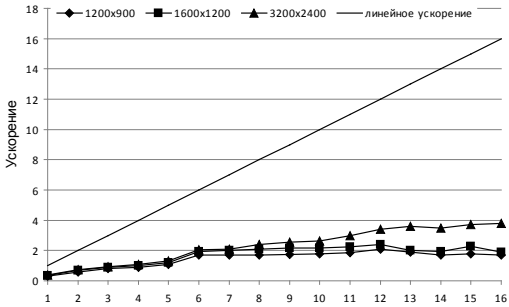


Рис. 5.

Выполнение параллельной программы на исследовательском кластере подтвердило наши предположения. Благодаря более мощным аппаратным средствам и архитектуре программного обеспечения мы получили более высокие значения ускорения (рис. 5).

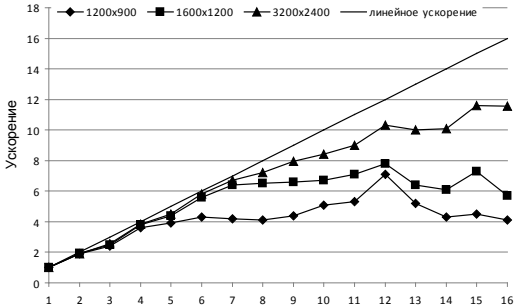


Рис. 6.

На исследовательском кластере использовалось до 16 узлов и полученные результаты сопоставимы с результатами лабораторного кластера (рис. 5 и 6). Здесь также наблюдался процесс насыщения ускорения, который для изображений с небольшим разрешением наблюдался раньше, чем для изображений с бóльшим разрешением.

Очевидны некоторые локальные оптимумы, которые наблюдаемые после 12 и 15 процессорами, в которых данные изображения имеют наилучшее распределение между вычислительными узлами, что приводит к оптимальному балансировке нагрузки. Однако детальный анализ этих оптимумов – тема дальнейшего исследования.

Выводы

Как побочный эффект исследования в работе представлен вариант реализации параллельного моделирования нейронных сетей с помощью

кластерных систем, который может быть использован в качестве базовой архитектуры. С развитием более быстрых и дешевых кластеров и сетевых интерфейсов может быть получено наилучшее соотношение цена/производительность для многих приложений, которые раньше считались прерогативой исключительно суперкомпьютеров, как в нашем случае при моделировании больших клеточных нейронных сетей.

Список использованной литературы

1. T. Roska, J. Vandewalle. Cellular Neural Network. John Wiley and Sons, 1993.
2. E. Schkuta, T. Fuerle, H. Wanec. Strutural data parallel simulation of neural networks. System Research and Info. Systems, 9:149-172, 2000.
3. Руденко О.Г., Бодянский Е.В. Основы теории искусственных нейроных сетей. – Харьков, НТУ «ХПИ», 2002. – 320 с.
4. Li Guo. The application of modified neural network ART1 in the fault diagnosis of air engine [J]. Microcompuer information, 2005(9-1):156-158.
5. V. Cimagalli and M. Balsi. Cellular neural networks: A review. In 6th Italian Workshop on Parallel Architectures and Neural Networks, Vietri sul Mare, May 1993. World Scientific.
6. E. Schikuta. Data parallel software simulation of cellular neural networks. In CNNA’96 - 1996 Fourth IEEE Internationl Workshop on Cellular neural networks and their applications, pages 267-271, 1996.
7. Таненбаум Э. Архитектура компьютера. – М, «Питер», 2005. – с. 581.
8. J. Ortega, R. Voigt. Solution of partial differential equationson vector and parallel computers. Society for Industrial& Applied Mathematics, July 1985.
9. http://www.zn.ua/3000/3100/60893/
10. http://www.comsys.ntu-kpi.kiev.ua