Mathematical Problems of Computer Science 23, 2004, 5–11.

On the Construction of Cluster Systolic Arrays*

Edmon M. Davtyan

Institue for Informatics and Automation Problems of NAS of RA
e-mail edmon@ipia.sci.am

Abstract

The paper presents three approaches to the construction of *cluster systolic arrays*. A careful analysis of these approaches based on the comparison of the running times of corresponding systolic arrays was carried out. A method to minimize the running time is proposed.

1 Introduction

A new approach to the modeling of one-dimensional systolic arrays of n cells on a cluster of $m \ll n$ processors is presented. The new approach is initially based on the mapping of the given systolic algorithm graph onto the cluster processors. As a result, a one-dimensional systolic array of m cells is constructed, which holds to be analogous to the given array from the standpoint of functioning of the array.

Consider now the class of one-dimensional systolic arrays of n cells, whose every non-boundary cell is interconnected according to Von Neumenn's or Moore's classical concept on 1-radius two-way neighborhood and each boundary cell is interconnected according to the same concept on one-way neighborhood [1]. Further we'll consider a systolic array in this class and denote it by $\mathcal{A}$.

Let $A = \{a_{ij}\}_{1 \le i \le s,\ 1 \le j \le n}$ be the table which describes the algorithm graph of the systolic array $\mathcal{A}$ taking s steps and let C be a cluster of m processors. The elements in the table A are connected by the relation $\prec$, so we may denote the algorithm graph of the systolic array $\mathcal{A}$ by $(A, \prec)$. A cluster-based implementation of the systolic algorithm requires that the work of corresponding one-dimensional systolic array given by table A be distributed among the cluster processors.

Consider now the mapping $\varphi : A \to \{1, 2, \ldots, s_\varphi\} \times \{1, 2, \ldots, m\}$, where $1 \le s_\varphi \le s$. For $\forall\, q, q' \in \{1, 2, \ldots, s_\varphi\},\ p, p' \in \{1, 2, \ldots, m\}$ if $q \ne q' \ \vee\ p \ne p'$, then the following conditions hold:

- $|\varphi^{-1}(q, p)| = |\varphi^{-1}(q', p')|$,
- $\varphi^{-1}(q, p)$ and $\varphi^{-1}(q', p')$ are isomorphic with respect to relation $\prec$.

Let $\varphi(A)$ denote the $s_\varphi \times m$-dimensional table obtained from partitioning of table A. If table $\varphi(A)$ is a homomorphic image of table A with respect to relation $\prec$, then it presents the work of a systolic array of m cells. The k-th column in the table $\varphi(A)$ describes the work

*This research is supported by INTAS - 0447, ISTC - 823 grants and 04.10.31 Target Program of RA.

of the k-th cell ($1 \le k \le m$) in this array and the k-th cell performs the work described in the t-th row of the table ($1 \le t \le s_\varphi$) at the t-th moment of time. We call the systolic array obtained by φ mapping *cluster systolic array* and denote it by A_φ. $\varphi(A)$ is the table which describes the graph of the systolic algorithm that corresponds to A_φ. For a cluster-based implementation of the systolic array of m cells, a one-to-one correspondence should be drawn between the total work of a cell in the array and each processor of the cluster.

Let R be the set of $s' \times n'$-dimensional tables, where $s' < s \wedge n' < n$. For any $r \in R$, if $a \in r$, then $a \in A$ and the elements in the table r are connected by the relation $\prec$. As an counter example, consider the mapping φ, where for any $q \in \{1, 2, \ldots, s_\varphi\}$ and $p \in \{1, 2, \ldots, m\}$, $\varphi^{-1}(q, p) \in R$. It is evident that the condition of homomorphism is violated here. Hence, this mapping doesn't allow construction of cluster systolic array.

To set the problem, we proceed with some additional conditions and notations that can be helpful to characterize the cluster and the systolic array. To describe a homogenous cluster C of m identical processors provided to implement the systolic array $\mathcal{A}$ of n cells, we'll use the system $< m, V_0, \delta_0, n, V_l, V_r, V_v, \Delta >$, where

1) V_0 is the maximal data size that can be transmitted between the processors of the cluster in a session within the required transmission time δ_0;

2) V_l and V_r are the data sizes of the cells in $\mathcal{A}$ systolic array transmitted in the left and right directions, respectively;

3) V_v is the total size of local variables in the systolic array $\mathcal{A}$ (it varies depending on the type of cluster processor);

4) Δ is the time required to perform the work of every cell in the systolic array $\mathcal{A}$ on a cluster processor (it depends on the way of software implementation and type of cluster processor).

Let C be a cluster, where the conditions $V_l + V_v \le V_0$ and $V_r + V_v \le V_0$ are satisfied, and let $\delta(V)$ be the time required to transmit data of V size between the nodes of the cluster, defined by the equation:

$$\delta(V) = \begin{cases} \delta_0 & , \text{ if } V \le V_0 \\ \delta_0 + \delta(V - V_0) & , \text{ if } V > V_0 \end{cases}.$$

Then, it is clear that the running time of the cluster systolic array A_φ of m cells can be determined by the formula:

$$t_\varphi = s_\varphi(|\varphi^{-1}(q, p)|\Delta + \delta(V_\varphi)),$$

where s_φ is the number of steps in the algorithm, $q \in \{1, 2, \ldots, s_\varphi\}$, $p \in \{1, 2, \ldots, m\}$ and V_φ is the data size transmitted between the steps.

Now we can set the problem.

The *problem of C cluster-based implementation of the systolic array $\mathcal{A}$* is : given a $(A, \prec)$, to find such a mapping $\varphi_0 : A \to \{1, 2, \ldots, s_{\varphi_0}\} \times \{1, 2, \ldots, m\}$ that $t_{\varphi_0} \le t_\varphi$ for an arbitrary cluster systolic array A_φ.

2 Comparison of cluster systolic arrays

Let $\mathcal{A}$ be a systolic array. In this section we'll try to analyze three approaches to work distribution of the systolic array $\mathcal{A}$ on cluster C. To this end we'll estimate the length of running times of obtained cluster systolic arrays and try to draw a comparison of them.

Define the numbers $x > 2$ and m':

$$x = \begin{cases} \left[\frac{n}{m}\right] & ,\text{if } n \bmod m = 0 \wedge \left[\frac{n}{m}\right] \bmod 2 = 0 \\ \left[\frac{n}{m}\right] + 1 & ,\text{if } \left[\frac{n}{m}\right] \bmod 2 \neq 0 \\ \left[\frac{n}{m}\right] + 2 & ,\text{if } n \bmod m \neq 0 \wedge \left[\frac{n}{m}\right] \bmod 2 = 0 \end{cases},$$

$$m' = \begin{cases} \left[\frac{n}{x}\right] & ,\text{if } n \bmod x = 0 \\ \left[\frac{n}{x}\right] + 1 & ,\text{otherwise} \end{cases},$$

where $m' \leq m$ gives the actual number of processors in the cluster among which the work of the systolic array should be distributed. Consequently the number of cells in the cluster systolic array is m'.

In the formulas below, the time required to transmit data of size V between the cluster nodes will be defined as follows:

$$\delta(V) = \begin{cases} \delta_0 & ,\text{ if } V \leq V_0 \\ \delta_0 + L_0(V - V_0) & ,\text{ if } V > V_0 \end{cases},$$

where δ_0, L_0 and V_0 are constants defined by the structure of the cluster.

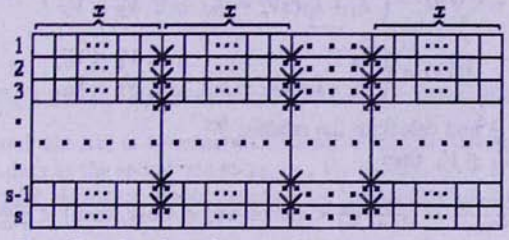

Fig.1 Systolic algorithm graph partition by "rows"

1°. Consider a very simple mapping $\varphi_1 : A \to \{1, 2, \ldots, s_{\varphi_1}\} \times \{1, 2, \ldots, m'\}$. We call the table $\varphi_1(A)$ a partition by x-length "rows". Fig.1 gives the graph partition of the systolic algorithm corresponding to $\mathcal{A}$.

The number of steps ($s_{\varphi_1} = s$) in the algorithm remains unchanged and every cluster processor sequentially performs the work of neighbor cells $|\varphi_1^{-1}(q, p)| = x$ in number of the systolic array $\mathcal{A}$. Every step of the algorithm ends in an interchange of data of V_l and V_r sizes between the neighbor processors.

Further calculate the running time of the cluster systolic array A_{φ_1} on a cluster of m' processors:

$$t_{\varphi_1} = s(x\Delta + 2\,\delta(V_l) + 2\,\delta(V_r)) = s(x\Delta + 4\,\delta_0).$$

2°. Now consider the mapping $\varphi_2 : A \to \{1, 2, \ldots, s_{\varphi_2}\} \times \{1, 2, \ldots, m'\}$, which was constructed by flow event structures (FES) [2] in [3]. We call the table $\varphi_2(A)$ a partition by "squares". A fragment of this partition is given in Fig.2 ($x = 4$). In this case every cluster processor sequentially performs the work of a group of cells in the systolic array $\mathcal{A}$ at different steps, which is actually the work of one cell in amount of $|\varphi_2^{-1}(q, p)| = \frac{x^2}{2}$. The number of steps is defined by the formula:

$$s_{\varphi_2} = \begin{cases} \frac{2s}{x} + 1 & ,\text{ if } s \bmod \frac{x}{2} = 0 \vee s \bmod \frac{x}{2} = 1 \\ \frac{2s}{x} + 2 & ,\text{ otherwise} \end{cases}.$$

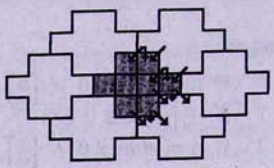

Fig.2 Systolic algorithm graph partition by "squares"

Suppose that we start counting the steps with 0. Every even step of the algorithm ends by an exchange of data of the size $V_R = xV_r + \frac{x}{2}V_v$ and every odd step ends by an exchange of data of the size $V_L = xV_l + \frac{x}{2}V_v$ between the processors. Denote $V_r' = V_r + \frac{1}{2}V_v$ and $V_l' = V_l + \frac{1}{2}V_v$. In this case $V_R = xV_r'$ and $V_L = xV_l'$. The running time of the cluster systolic array A_{φ_2} is described by the formula:

$$t_{\varphi_2} = \frac{s_{\varphi_2}}{2}\left(\frac{x^2}{2}\Delta + 2\,\delta(V_R)\right) + \frac{s_{\varphi_2}}{2}\left(\frac{x^2}{2}\Delta + 2\,\delta(V_L)\right) = s_{\varphi_2}\left(\frac{x^2}{2}\Delta + \delta(V_R) + \delta(V_L)\right).$$

$\delta(V_R)$ and $\delta(V_L)$ are defined as:

$$\delta(V_R) = \begin{cases} \delta_0 & , \text{ if } V_R \le V_0 \\ \delta_0 + L_0(xV_r' - V_0) & , \text{ if } V_R > V_0 \end{cases},$$

$$\delta(V_L) = \begin{cases} \delta_0 & , \text{ if } V_L \le V_0 \\ \delta_0 + L_0(xV_l' - V_0) & , \text{ if } V_L > V_0 \end{cases}.$$

Assume that $s_{\varphi_2} = \frac{2s}{x}$ and calculate the relation $\frac{t_{\varphi_2}}{t_{\varphi_1}}$.

If $V_R \le V_0$ and $V_L \le V_0$, then

$$\frac{t_{\varphi_2}}{t_{\varphi_1}} = \frac{\frac{2s}{x}(\frac{x^2}{2}\Delta + 2\ \delta_0)}{s(x\Delta + 4\ \delta_0)} = \frac{x^2\Delta + 4\ \delta_0}{x^2\Delta + 4\ \delta_0 x} < \{\text{since } 4\ \delta_0 < 4\ \delta_0 x, \text{ then}\} < \frac{x^2\Delta + 4\ \delta_0 x}{x^2\Delta + 4\ \delta_0 x} = 1.$$

If $V_R > V_0$ and $V_L \le V_0$, then

$$\frac{t_{\varphi_2}}{t_{\varphi_1}} = \frac{\frac{2s}{x}(\frac{x^2}{2}\Delta + 2\ \delta_0 + L_0(xV_r' - V_0))}{s(x\Delta + 4\ \delta_0)} = \frac{x^2\Delta + 4\ \delta_0 - 2\ L_0V_0 + 2\ L_0V_r'x}{x^2\Delta + 4\ \delta_0 x} = \{\text{since } V_r' \le V_0 \text{ and}$$
$$L_0V_0 = L_0V_r' = \delta_0, \text{ then}\} = \frac{x^2\Delta + 2\ \delta_0 + 2\ \delta_0 x}{x^2\Delta + 4\ \delta_0 x} = \frac{x^2\Delta + 2\ \delta_0(1+x)}{x^2\Delta + 4\ \delta_0 x} < \{\text{since } 2(1+x) < 4x,$$
$$\text{then}\} < \frac{x^2\Delta + 4\ \delta_0 x}{x^2\Delta + 4\ \delta_0 x} = 1.$$

If $V_R \le V_0$ and $V_L > V_0$, then similarly, we have $\frac{t_{\varphi_2}}{t_{\varphi_1}} < 1$.

If $V_R > V_0$ and $V_L > V_0$, then

$$\frac{t_{\varphi_2}}{t_{\varphi_1}} = \frac{\frac{2s}{x}(\frac{x^2}{2}\Delta + 2\ \delta_0 + L_0(xV_r' + xV_l' - 2\ V_0))}{s(x\Delta + 4\ \delta_0)} = \frac{x^2\Delta + 4\ \delta_0 - 4\ L_0V_0 + 2(L_0V_r' + L_0V_l')x}{x^2\Delta + 4\ \delta_0 x} = \{\text{since } V_r' \le V_0\,,\, V_l' \le V_0$$
and $L_0V_0 = L_0V_r' = L_0V_l' = \delta_0$, then$\} = \frac{x^2\Delta + 2\ \delta_0 x}{x^2\Delta + 4\ \delta_0 x} < \frac{x^2\Delta + 4\ \delta_0 x}{x^2\Delta + 4\ \delta_0 x} = 1$ as $2\ \delta_0 x < 4\ \delta_0 x$:

These considerations imply that in general case

$$t_{\varphi_2} < t_{\varphi_1}.$$

3°. Consider the mapping $\varphi_3 : A \to \{1, 2, \ldots, s_{\varphi_3}\} \times \{1, 2, \ldots, m'\}$ and the table $\varphi_3(A)$ of graph partition of the given systolic algorithm, which is obtained from partitioning by $\lambda \ge 2$-length and $\omega \ge 2$-width "rectangles", where $\lambda + \omega = x$ and either $\lambda < \frac{4}{5}x \Rightarrow \omega > \frac{1}{5}x$ or $\omega < \frac{4}{5}x \Rightarrow \lambda > \frac{1}{5}x$. In this case as well every cluster processor sequentially performs the

work of a group of cells in systolic array $\mathcal{A}$ at different steps, which is in fact the work of one cell in amount of $|\varphi_3^{-1}(q,p)| = 2\lambda\omega$. If we try to give a lower-bounded estimate of the number of "rectangles" in the algorithm graph, then we have this number equal to $\frac{sx}{2\lambda\omega}$. The number of cells in the modified systolic array will be $\frac{n}{\lambda+\omega}$, hence for this case the number of steps is characterized as $s_{\varphi_3} = \frac{s(\lambda+\omega)}{2\lambda\omega}$.

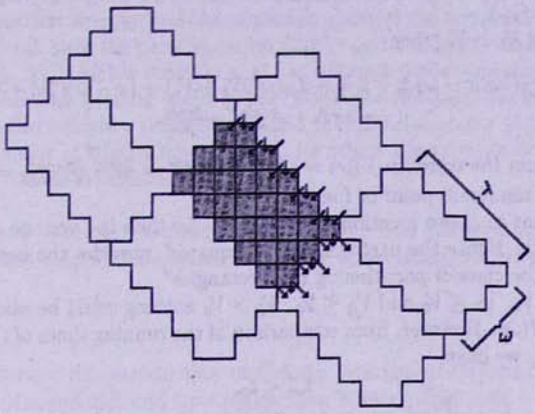

Fig.3 Systolic algorithm graph partition by "rectangles"

In this case as well the size of transmitted data at the end of even steps differs from the size of transmitted data at the end of odd steps, i.e., $V_R = \omega(2V_r + V_v)$ and $V_L = \lambda(2V_l + V_v)$. A fragment of systolic algorithm graph partition by "rectangles" is given in Fig.3, where $\lambda = 5$ and $\omega = 3$.

It is easy to see that the case $\lambda = \omega$ matches with the systolic algorithm graph partition by "squares". Estimate the running time of the algorithm on a cluster of m' processors for the case of partitioning by "rectangles":

$$t_{\varphi_3} = s_{\varphi_3}(2\lambda\omega\Delta + \delta(V_R) + \delta(V_L)).$$

Now examine the function $f(\omega)$ that characterizes the running time of the algorithm. Our desire is to find for what relationship between λ and ω the running time stand for the minimum? This examination will be also helpful to compare the current and the prior approaches.

Since $\lambda = x - \omega$, then

$$f(\omega) = \frac{sx}{2\omega(x-\omega)}(2\omega(x-\omega)\Delta + \delta(V_R) + \delta(V_L)).$$

$\delta(V_R)$ and $\delta(V_L)$ are defined as:

$$\delta(V_R) = \begin{cases} \delta_0 & , \text{ if } V_R \le V_0 \\ \delta_0 + L_0(\omega(2V_r + V_v) - V_0) & , \text{ if } V_R > V_0 \end{cases},$$

$$\delta(V_L) = \begin{cases} \delta_0 & , \text{ if } V_L \le V_0 \\ \delta_0 + L_0((x-\omega)(2V_l + V_v) - V_0) & , \text{ if } V_L > V_0 \end{cases}.$$

If $V_R \le V_0$ and $V_L \le V_0$, then

$$f(\omega) = \frac{sx}{2\omega(x-\omega)}(2\omega(x-\omega)\Delta + 2\,\delta_0) = sx\Delta + \frac{sx\delta_0}{x\omega - \omega^2}.$$

From the equality $f'(\omega) = -sx\delta_0\frac{x-2\omega}{(x\omega-\omega^2)^2} = 0$ we obtain that $\omega = \frac{x}{2}$ is the minimum point of function $f(\omega)$.

If $V_R > V_0$ and $V_L > V_0$, then

$$f(\omega) = \frac{sx}{2\omega(x-\omega)}(2\omega(x-\omega)\Delta + 2\,\delta_0 + L_0(\omega(2V_r + V_v) + (x-\omega)(2V_l + V_v) - 2V_0)) = \\ = sx\Delta + \frac{3sx\delta_0}{2(x-\omega)} + \frac{3sx\delta_0}{2\omega}.$$

In this case from the equality $f'(\omega) = \frac{3sx\delta_0}{2(x-\omega)^2} + \frac{3sx\delta_0}{2\omega^2} = \frac{3sx\delta_0}{2}\frac{2x\omega - x^2}{(x-\omega)^2\omega^2} = 0$ we also obtain that $\omega = \frac{x}{2}$ is the minimum point of function $f(\omega)$.

This means that in above mentioned cases if $\lambda = \omega$, then the systolic array A_{φ_2} has the best operating rate. Hence the partitioning by "squares" provides the best running time of the algorithm in the class of partitioning by "rectangles".

In cases $V_R > V_0$, $V_L \le V_0$ and $V_R \le V_0$, $V_L > V_0$ nothing could be said about minimum point of function $f(\omega)$. However, from comparison of the running times of the cluster systolic array A_{φ_2} with t_{φ_2} we have

$$t_{\varphi_2} < t_{\varphi_2}.$$

The conclusions drawn here and the fact that φ_2 provides the best speedup [3] make some background to state that the proposed mapping φ_2 is an effective approach to construct cluster systolic arrays.

3 Minimization of the running time

Now consider the function $g(x)$ that characterizes the running time of the cluster systolic array A_{φ_2}, where

$$g(x) = s_{\varphi_2}(\frac{x^2}{2}\Delta + \delta(V_R) + \delta(V_L)).$$

Examine the function $g(x)$.

If $V_R \le V_0$, $V_L \le V_0$ and $s_{\varphi_2} = \frac{2s}{x} + 1$, then

$$g(x) = s\Delta x + \frac{4s\delta_0}{x} + \frac{x^2}{2}\Delta + 2\delta_0.$$

Calculate the derivative of the function:

$$g'(x) = s\Delta - \frac{4s\delta_0}{x^2} + \Delta x.$$

¿From the equality $g'(x) = 0$ we obtain the cubic equation $x^3 + sx^2 - \frac{4s\delta_0}{\Delta} = 0$. By solving this equation we may find the minimum point of the function.

When $s_{\varphi_2} = \frac{2s}{x} + 2$, we obtain the cubic equation $2x^3 + sx^2 - \frac{4s\delta_0}{\Delta} = 0$.

If $V_R > V_0$, $V_L \le V_0$ or $V_R \le V_0$, $V_L > V_0$, and $s_{\varphi_2} = \frac{2s}{x} + 1$, then

$$g(x) = s\Delta x + \frac{2s\delta_0}{x} + 2s\delta_0 + \frac{x^2}{2}\Delta + \delta_0 x + \delta_0.$$

Calculate the derivative of the function:

$$g'(x) = s\Delta - \frac{2s\delta_0}{x^2} + \Delta x + \delta_0.$$

¿From the equality $g'(x) = 0$ we obtain the cubic equation $\Delta x^3 + (s\Delta + \delta_0)x^2 - 2s\delta_0 = 0$. By solving this equation we may find the minimum point of the function.

When $s_{\varphi_2} = \frac{2s}{x} + 2$, then the cubic equation $2\Delta x^3 + (s\Delta + 2\,\delta_0)x^2 - 2s\delta_0 = 0$ is obtained.

In case $V_R > V_0$, $V_L > V_0$ the function $g(x)$ has a unique extremum point with a negative value. Hence for this case nothing could be said about the minimum point of the function.

In cases, when there exists a possibility to find the minimum point of function $g(x)$, one may define the number of cluster processors m' for which the running time of the systolic algorithm $\mathcal{A}$ on a cluster is minimal.

Acknowledgments. The author wishes to thank Prof. Yuri Shoukourian and Dr. Karine Shahbazyan for many helpful discussions, comments, and suggestions regarding this work.

References

[1] Marianne Delorme. An introduction to Cellular Automata. Cellular Automata: a Parallel Model, Mathematics and Its Applications, Kluwer, July 1998.

[2] K.V.Shahbazyan, Yu.H.Shoukourian. Logically Definable Languages of Computations in one Class of Flow Event Structures. INTAS grant "Weak Arithmetics" 2000-447, (2002).

[3] E.Davtyan. On the Modelling of One Class of Systolic Structures on a PC Cluster. In proceedings of CSIT-2003, pp. 340-344.

Կլաստերային սիստոլիկ զանգվածների կառուցման մասին

Է. Մ. Դավթյան

Ամփոփում

Աշխատանքում դիտարկված է *կլաստերային սիստոլիկ զանգվածների* կառուցման երեք մոտեցում: Կատարված է այդ մոտեցումների համեմատություն ըստ համապատասխան սիստոլիկ զանգվածների աշխատանքների կատարման ժամանակների: Առաջարկված է աշխատանքի կատարման ժամանակը մինիմիզացնող մեթոդ: