индекс 3624

ЕРЕВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ

Препринт ЕФИ-1123(86)-88

ԵՐԵՎԱՆԻ ՖԻԶԻԿԱՅԻ ԻՆՍՏԻՏՈՒՏ
ЕРЕВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ
YEREVAN PHYSICS INSTITUTE

М.А.МХИТАРЯН, А.Б.НЕРСЕСЯН

ПАРАЛЛЕЛЬНО-ПОТОЧНАЯ ИНТЕРПРЕТАЦИЯ ОБРАЩЕНИЯ МАТРИЦЫ МЕТОДОМ ГАУССА-ЖОРДАНА

ЦНИИатоминформ
ЕРЕВАН—1988

Նախնատիպ ԵՖԻ-II23(86)-88

Մ.Ա.ՄԽԻԹԱՐՅԱՆ, Ա.Բ.ՆԵՐՍԻՍՅԱՆ

ՄԱՏՐԻՑԱՅԻ ՀՈՒԳԱՀԵՌ-ՀՈՍՈՒՆ ԴԱՐՁՄԱՆ ԲԱՑԱՏՐՈՒԹՅՈՒՆԸ ԳԱՈՒՍ-ԺՈՐԴԱՆԻ ՄԵԹՈԴՈՎ

Աշխատանքում քննարկված է մատրիցայի դարձման հեղհեղային հաշվման սխեմը Գաուս-Ժորդանի մեթոդով՝ հաշվարկման իրացման տարածա-ժամանակային առանձնահատկությունների հաշվառումով: Հաշվված է արդյունավետության գործակիցը, հետազոտվում են նրա բարձրացման հնարավորությունները, գնահատվում է պրոցեսորների տվյալ պահին աշխատող առավելագույն թիվը:

Երևանի Ֆիզիկայի ինստիտուտ

Երևան 1988

Препринт ЕФИ-II23(86)-88

УДК 681.3.012:519.67

М.А.МХИТАРЯН, А.Б.НЕРСЕСЯН

ПАРАЛЛЕЛЬНО-ПОТОЧНАЯ ИНТЕРПРЕТАЦИЯ ОБРАЩЕНИЯ МАТРИЦЫ МЕТОДОМ ГАУССА-ЖОРДАНА

В работе рассматривается потоковая вычислительная схема обращения матрицы методом Гаусса-Жордана с учетом пространственно-временных особенностей реализации вычислений. Вычисляется коэффициент эффективности, исследуются возможности его повышения, оценивается максимальное число работающих в данный момент времени процессоров.

Ереванский физический институт

Ереван 1988

Preprint YERPHI-1123(86)-88

M.A. MKHITARYAN, .B. NERSESSYAN

PARALLEL-FLOW INTERPRETATION OF MATRIX INVERSION BY GAUSS-JORDAN METHOD

A flow computational scheme for matrix inversion by the Gauss-Jordan method with account of space-time peculiarities of computation realization has been considered. The efficiency is calculated, possibilities for its improvement are studied, a maximum number of operating processors is estimated.

Yerevan Physics Institute
Yerevan 1988

Параллельный алгоритм обращения матрицы посредством компактной схемы Гаусса с учетом пространственно-временных особенностей реализации вычислений рассматривался в работе С.Г. Седухина [I] , где была предложена соответствующая вычислительная схема, представляющая собой сеть ортогонально связанных процессоров с требуемой коммуникацией. Эквивалентным упомянутому алгоритму по количеству необходимых для вычисления операций является алгоритм Гаусса-Жордана для обращения матрицы [2]. В настоящей работе исследуется последний вариант нахождения обратной матрицы, рассматривается потоковая схема для реализации параллельного алгоритма Гаусса-Жордана, учитывающая пространственно-временную структуру используемых преобразований.

Дана неособенная $[n \times n]$ матрица $A=[a_{ij}]$. Добавив к ней единичную $[n \times n]$ матрицу $E=\delta_{ij}$ - получим расширенную $[n \times 2n]$ матрицу $[A,E]=[\beta_{ij}]$, где $\beta_{ij}=a_{ij}$ при $1 \leq j \leq n$; $\beta_{ij}=\delta_{ij}$ при $n+1 < j \leq 2n$.

После применения к ней метода исключения Гаусса-Жордана матрица примет вид $[E,A^{-1}]$, где A^{-1} - обратная матрица. Данный процесс можно описать следующим образом:

$$a_{ij}^{(k)} = \text{if } i=j=k \text{ then } 1$$

$$\text{if } i=k \text{ then } a_{kj}^{(k-1)} / a_{kk}^{(k-1)}$$

$$\text{if } j=k \text{ then } 0$$

$$\left\{ a_{ij}^{(k-1)} - a_{ik}^{(k-1)} \cdot a_{kj}^{(k-1)} / a_{kk}^{(k-1)} \right\}$$

$$\delta_{ij}^{(k)} = \text{if } i=j=k \text{ then } 1/a_{kk}^{(k-1)}$$

$$\text{if } i=k>j \text{ then } \delta_{kj}^{(k-1)} / a_{kk}^{(k-1)}$$

$$\text{if } i=k<j \text{ then } 0$$

$$\text{if } j=k \text{ then } -a_{ik}^{(k-1)} / a_{kk}^{(k-1)}$$

$\left\{ \delta_{ij}^{(k-1)} - a_{ik}^{(k-1)} \cdot \frac{\delta_{kj}^{(k-1)}}{a_{kk}^{(k-1)}} \right\}$, где k – номер ведущей строки $(k=\overline{1,n})$.

Каждый этап вычислений начинается тогда, когда элементы ведущей строки делятся на соответствующий диагональный элемент $a_{kk}^{(k-1)}$, т.е. ведущая строка нормализуется. Индекс $(k-1)$ сверху указывает на число предшествующих этапов.

Данный процесс реализуется на сети из $p \leq n^2$ ортогонально связанных процессоров. Рассмотрим несколько типичных случаев.

I. Число вычислителей равно числу элементов матрицы A, $p=n^2$. Мы действуем на ортогональной сети размерности $(n \times n)$.

Процесс начинается возбуждением вычислителя (I,I) и передачей им операнда a_{11}^{-1} процессорам (I,2) и (2,1). На втором шаге выполняются следующие действия: (I,2) вычисляет $a_{12} \cdot a_{11}^{-1}$, передает a_{11}^{-1} процессору (I,3), а вычисленное произведение

транслируется (2,2), тем временем (2,1) выполняет $(-a_{21} \cdot a_{11}^{-1})$ и передает a_{11}^{-1} в (3,I). Таким образом, на третьем шаге первого этапа будут работать процессоры (I,3), (2,2) и (3,I). Сигнал, распространяясь по сети ортогонально связанных процессоров, последовательно активизирует их.

На k-ом этапе работа (i,j) – вычислителя должна состоять из последовательного преобразования элементов a_{ij}^{k-1} и $\delta_{ij}^{(k-1)}$. Но, учитывая специфику вышеописанного алгоритма, работа (i,j) процессора несколько упрощается: при $j \leq k$ вычисляется только $\delta_{ij}^{(k-1)}$, а при $j>k$ $a_{ij}^{(k-1)}$.

Естественно, все это происходит после вычисления элемента $a_{kk}^{(k-1)}$ (k,k) – процессором, после чего последний передает его соседним вычислителям, готовым к его приему. А сам (k,k) процессор считает обратную величину $[a_{kk}^{(k-1)}]^{-1}$.

Обмен информацией между соседними вычислителями осуществляется с помощью регистров R_1 и R_2 (рис.I). Посредством регистра R_1 операнды распространяются двухсторонне в вертикальном направлении, а с помощью R_2 – в горизонтально (также двухсторонне). Этот процесс локален, т.е. обмен происходит только между соседними процессорами по мере готовности их регистров к обмену.

Вычислим теперь время обращения матрицы A. За один такт, т.е. за единицу времени работы вычислителя, принимается время выполнения операции накопления $\alpha \longleftarrow \alpha + a \cdot b$.

Элементы $a_{kk}^{(k-1)}$ вычисляются на третьем шаге $(k-1)$ этапа, кроме a_{11}^{-1}. Следовательно, элемент $a_{nn}^{(n-1)}$ будет вычислен на $m=3(n-1)$ шаге. Время обращения матрицы A будет

равно $T = mt' + T'$, где t' - единица времени работы процессора ($t' = 1$), T' - время выполнения последнего этапа.

$$T' = \max_{\substack{1 \le i \le n \\ 1 \le j \le n}} [(n - i + n - j) + 1]$$

Выражение в круглых скобках указывает на время задержки получения (i,j) вычислителем операнда $a_{nn}^{(n-1)}$ в течение последнего этапа, а единица - время работы (i,j) - вычислителя. Последний этап заканчивается в (I,I) процессоре (рис.2). Следовательно,

$$T' = 3 \cdot (n-1) + 2n - 1 - 1 = 5n - 5.$$

Последнее вычитание единицы учитывает тот факт, что время вычисления элемента $a_{nn}^{(n-1)}$ входит дважды в T. Таким образом, коэффициент эффективности в данном случае будет равен $K_{эф} \approx \frac{I}{5} =$ $=0,2$, что выше коэффициента эффективности 0,I4 (см. I) при использовании n^2 ортогонально связанных процессоров при вычислении обратной матрицы по компактной схеме Гаусса.

Теперь оценим максимальное число активных процессоров. При последовательной и параллельной интерпретациях метода Гаусса-Жордана асимптотическая операционная сложность не изменяется:

$$\sum_{t=1}^{5n-5} P_t \approx n^3 \quad (n \to \infty),$$

где P_t - число работающих вычислителей в момент времени t

Заменив в сумме (I) P_t на P_{max}, будем иметь

$$n^2 \geqslant P_{max} \geqslant \frac{n^3}{5n-5} > \frac{n^2}{5}.$$

Максимальное число активных процессоров при реализации метода Гаусса-Жордана на сети n^2 ортогонально связанных вычислителей больше пятой части имеющегося их количества. На рис.3 изображен график зависимости числа активных процессоров P_t от времени t при $n = 4$.

2. Имеется $p = \frac{n^2}{2}$ процессоров, если n - четно, в противном случае (при n - нечетном) $p = \frac{n(n+1)}{2}$. Достаточно рассмотреть первый случай.

(i,j) процессор запоминает элементы $a_{i,2j-1}, a_{i,2j}$ ($1 \le i \le n$, $1 \le j \le \frac{n}{2}$) данной матрицы A и соответствующие элементы единичной матрицы E. Преобразования будут выполняться с помощью вычислительной схемы размерности $(n \times \frac{n}{2})$, включающей $\frac{n^2}{2}$ процессоров, аналогичной изображенной на рис.I. Переходы между элементами главной диагонали можно отнести к двум типам по следующему признаку.

К I типу - те переходы, при которых элементы $a_{2j-1,2j-1}$, $a_{2j,2j}$ запоминаются вычислителями одного столбца (при нечетном j, рис.4а), а к II - те, при которых элементы главной диагонали запоминаются процессорами соседних столбцов (при четном j, рис.4б). Естественно, аналогичная ситуация будет наблюдаться в течение всего вычислительного процесса.

Для I типа переходов, если вычислен элемент $a_{2j-1,2j-1}^{(2(j-1))}$, то следующий $a_{2j,2j}^{(2j-1)}$ будет подсчитан на четвертом шаге $2j$-го этапа. Количество переходов такого типа $\frac{n}{2} - 1$. (Вычисление элемента $a_{22}^{(1)}$ рассматривается отдельно, так как он, очевидно, вычисляется на третьем шаге первого этапа). Время, затраченное на вычисление $a_{22}^{(1)}$, $a_{44}^{(3)}$, ..., $a_{nn}^{(n-1)}$ равно

$$T' = 4 \cdot \left(\frac{n}{2} - 1\right) + 3 .$$

Для II типа переходов время вычисления $a_{33}^{(2)}$, $a_{55}^{(4)}$, ... , $a_{n-1,n-1}^{(n-2)}$

$$T'' = 3 \cdot \left(\frac{n}{2} - 1\right) .$$

Величина $a_{nn}^{(n-1)}$ будет вычислена по истечении времени

$$T^{*} = T' + T'' = \frac{7n}{2} - 4 .$$

Последний этап начинается при последнем возбуждении вычислителя $(n, \frac{n}{2})$ и передачей им операнда $a_{nn}^{(n-1)}$ процессорам $(n-1, \frac{n}{2})$ и $(n, \frac{n}{2}-1)$. Время вычисления последнего этапа (рис.5).

$$T^{**} = \max_{\substack{1 \le i \le n \\ 1 \le j \le n}} \left[(n - i + \frac{n}{2} - j) + 2 \right] = \frac{3n}{2} .$$

Выражение в круглых скобках указывает на время задержки получения (i, j) - вычислителем операндов, необходимых для дальнейшего действия, 2 - время работы самого процессора.

Таким образом, полное время обращения матрицы на сети ортогонально связанных процессоров будет равно

$$T = T^{*} + T^{**} = 5n - 4 .$$

Коэффициент эффективности в этом случае $K_{эф} \simeq 0{,}4$, что вдвое больше коэффициента эффективности при n^2 процессорах.

Максимальное число активных в момент времени t процессоров оценим с помощью асимптотического тождества типа (I)

$$\frac{n^2}{5} < P_{max} \leqslant \frac{n^2}{2} .$$

На рис.6 показан график зависимости количества активных процессоров от времени при работе вычислительной схемы, включающей $\frac{n^2}{2}$ ортогонально связанных вычислителей ($n = 4$).

3. Количество процессоров $p = \frac{n^2}{4}$. Разобьем исходную матрицу A на блоки размерности (2 x 2). Каждый вычислитель запоминает по четыре элемента соответствующего блока. Аналогично разбивается и единичная матрица E, (i, j) блок которой фиксируется в локальной памяти (i, j) - го процессора.

Здесь также выделим переходы двух видов. Переходы между главными диагональными элементами, находящимися в одном блоке (2 x 2), назовем переходами I-го вида, между соседними элементами главной диагонали, содержащихся в разных блоках - переходами 2-го вида (рис.7).

Каждый (i, j) блок является эпицентром двух этапов. $2i-1$ этап начинается после вычисления (i, i) блоком элемента $a_{2i-1,2i-1}^{(2\cdot(i-1))}$, транслируемого соседним вычислителям, готовым к его приему. А сам (i, i) - процессор вычисляет элементы (i, i) - блока на данном этапе. Если в (i, j) - блоке начался $(2i-1)$ этап, то $2i$ в нем начнется только тогда, когда вычислятся все элементы этого блока на $(2i-1)$ -ом этапе.

В предыдущих случаях вычисление (i, i) - процессором обратной величины $\left[a_{2i-1,2i-1}^{(2\cdot(i-1))} \right]^{-1}$ не задерживало начало $2i$ этапа. В настоящем же случае - наоборот, поскольку вычисления, необходимые для начала $2i$ этапа производит тот же (i, i) процессор. Порядок обработки элементов блока в процессоре зависит от места поступления возбуждающего сигнала (по отношению к ведущему диагональному вычислителю $(i = j = k)$) и определяется локально.

Время вычисления обратной матрицы A^{-1} на сети $\frac{n^2}{4}$ ортогонально связанных вычислителей

$$T = T^{*} + T^{**} = 6n - 6.$$

(Обозначения те же, что и в п.2, только с учетом различного числа процессоров)

$$T^{*} = T' + T'' = 5n - 6,$$

где соответственно $T' = 3 \cdot \left(\frac{n}{2} - 1\right)$, $T'' = 7 \cdot \left(\frac{n}{2} - 1\right) + 4$,а время реализации последнего этапа

$$T^{**} = \max_{\substack{1 \le i \le n/2 \\ 1 \le j \le n/2}} \left[\left(\frac{n}{2} - i + \frac{n}{2} - j\right) + 4\right] = n + 2.$$

Коэффициент эффективности будет равен $K_{эфф} = \frac{2}{3}$, что выше коэффициента эффективности в предыдущих двух случаях.

Оценка максимального числа процессоров, работающих в момент времени t , будет следующей

$$\frac{n^2}{6} < P_{max} \leqslant \frac{n^2}{4}.$$

На рис.8 представлен график зависимости количества активных процессоров от времени при работе вычислительной схемы, включающей $\frac{n^2}{4}$ процессоров, при $n = 4$.

Заключение

Таким образом, рассмотрев три случая и для каждого из них посчитав коэффициент эффективности, оценив максимальное число активных в данный момент времени процессоров, мы имеем возможность сравнить, насколько эффективно интерпретирован параллельный алгоритм Гаусса-Жордана для обращения матрицы при разном количестве вычислителей. Полученные результаты обобщены в таблице.

Как видно из таблицы, самый высокий коэффициент получается при использовании $\frac{n^2}{4}$ процессоров ($K_{эф} = 0{,}66$). Уменьшение числа процессоров без изменения схемы коммуникаций влекло за собой повышение коэффициента эффективности. В последнем рассмотренном случае в течение процесса вычислений максимальное число активных процессоров более трети времени достигало имеющегося их количества 4 (рис.8).

Разумеется, при обсуждении результатов таблицы надо учитывать, что увеличение коэффициента эффективности зафиксировано без учета усложнения функций процессоров при $P = \frac{n^2}{2}$ и $P = \frac{n^2}{4}$

Таблица

P	n^2	$n^2/2$	$n^2/4$
$K_{эф.}$	0.2	0.4	0.66
P_{max}	$\frac{n^2}{5} < P_{max} \leqslant n^2$	$\frac{n^2}{5} < P_{max} \leqslant \frac{n^2}{2}$	$\frac{n^2}{6} < P_{max} \leqslant \frac{n^2}{4}$

Тем не менее, даже с учетом этих технических моментов представляется предпочтительным разумное уменьшение размерности сети ортогонально связанных вычислителей по отношению к размерности матрицы.

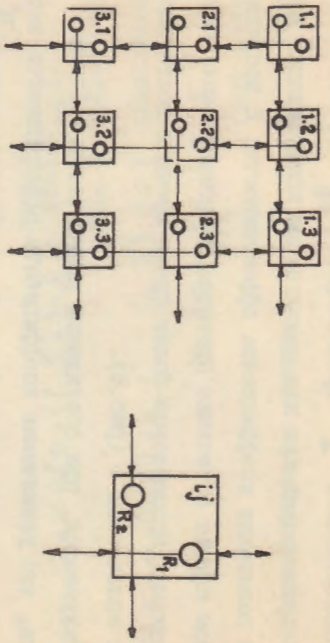

Рис.1

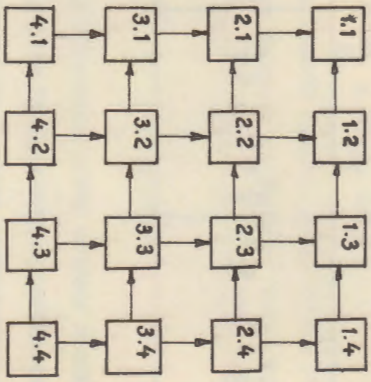

Рис.2

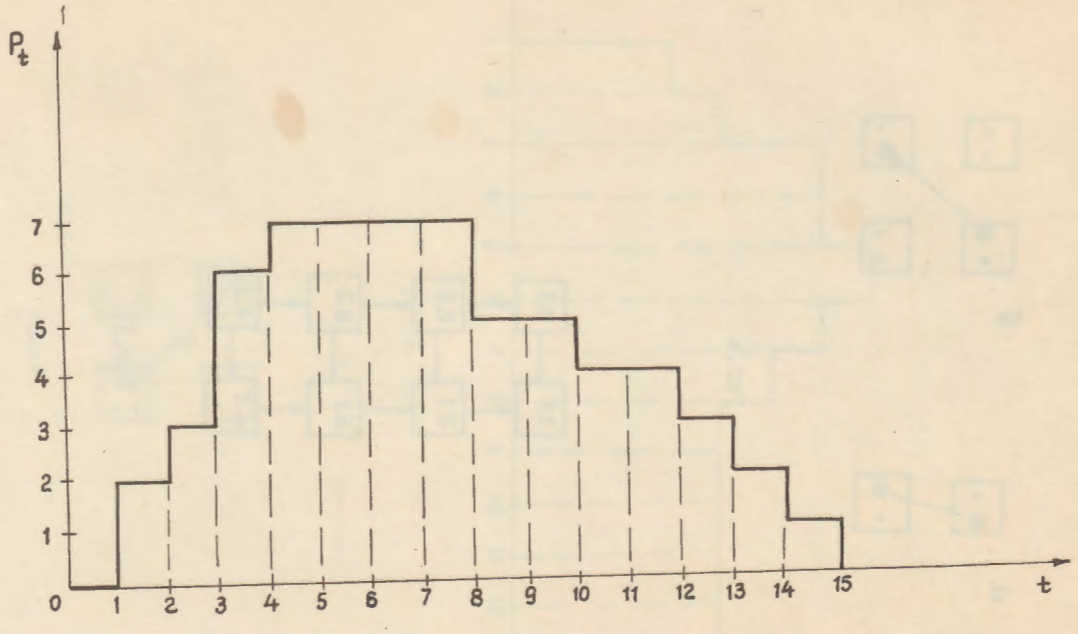

Рис.3

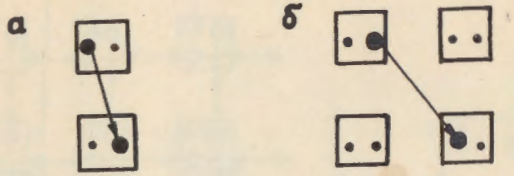

Рис.4

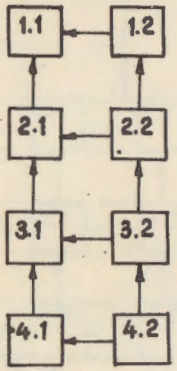

Рис.5

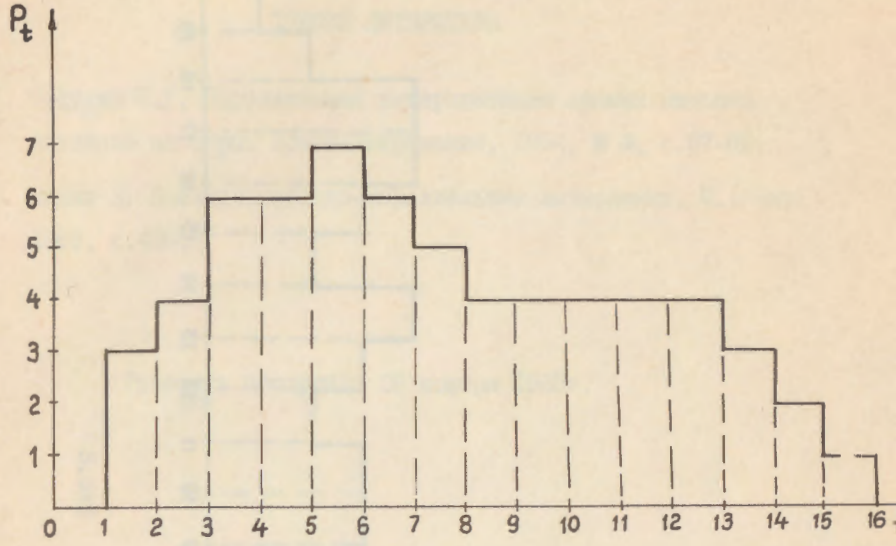

Рис.6

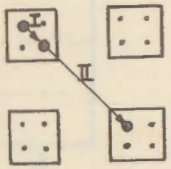

Рис.7

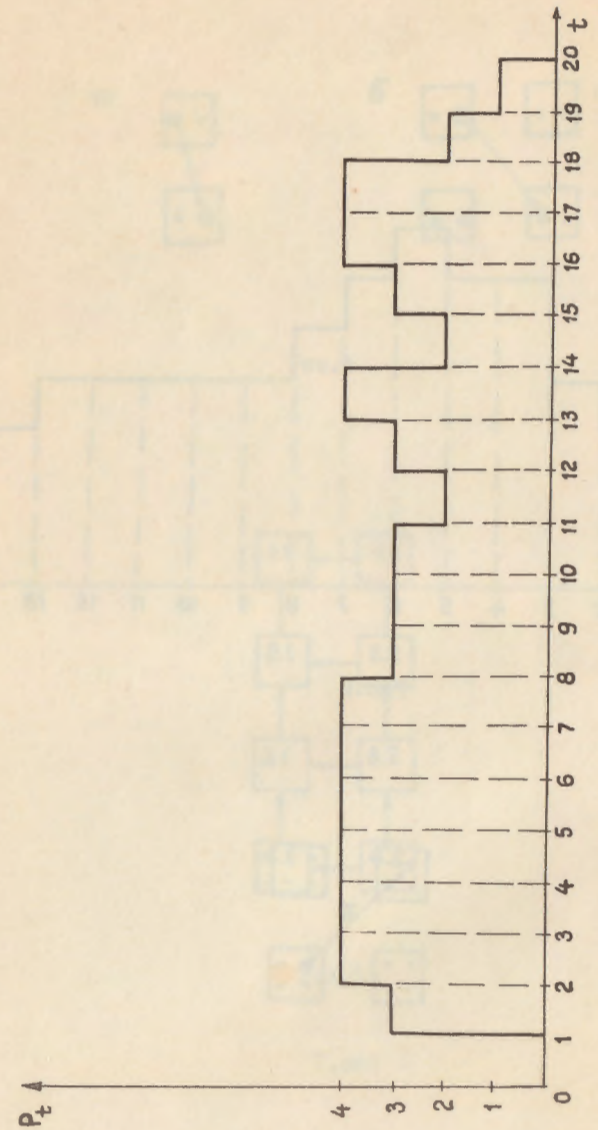

Рис.8

СПИСОК ЛИТЕРАТУРЫ

1. Седухин С.Г. Параллельная интерпретация прямых методов линейной алгебры. Программирование, 1984, № 4, с.57-68,

2. Валях Е. Последовательно-параллельные вычисления, М.: Мир, 1985, с.456.

Рукопись поступила 28 ноября 1988г.