ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Сборник трудов
1963 г. Института математики СО АН СССР Выпуск 9

К ВОПРОСУ ЭФФЕКТИВНОГО РЕШЕНИЯ ПРОСТЕЙШИХ ЗАДАЧ АЛГЕБРЫ МАТРИЦ НА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ (ВС), СОСТОЯЩЕЙ ИЗ L МАШИН

Г.А. Бекишев

В настоящей статье рассматриваются алгоритмы для решения основных задач на действия с матрицами, предусматривающие параллельное выполнение некоторых операций, т.е. алгоритмы, реализуемые на ВС. Исследуется вопрос об оптимальных алгоритмах этого рода. Допуская возможность параллельного выполнения операций при решении данной матричной задачи, мы ставим в статье вопрос о минимальном числе h шагов вычислений, за которое она может быть решена на основе соответствующего определения. В тесной связи с этим вопросом в статье рассматривается также вопрос о минимальном числе машин, которое потребуется для того, чтобы решить данную задачу за h шагов вычислений. В целом вся задача в соответствии со статьёй [I] формулируется как задача об оптимальном алгоритме некоторого их семейства $\mathcal{A}_L$, которое всякий раз описывается. В статье используется терминология и результаты, содержащиеся в работах [I], [2].

§ I. Сложение 2-х матриц. Умножение матрицы на число. Умножение прямоугольной матрицы на диагональную

Так как оценки оптимальных алгоритмов для решения указанных задач на ВС будут одинаковы, то мы ограничимся подроб-

ным рассмотрением одной из этих задач, например, задачи сложения 2-х матриц на ВС.

I.I. Сложение двух прямоугольных матриц. Пусть $\mathcal{A}_L = \{\mathcal{A}\}$ обозначает семейство всевозможных алгоритмов сложения 2-х прямоугольных матриц

$$A=(a_{ik}),\quad B=(b_{ik})$$

$$(i=1,2,\ldots,m;\quad k=1,2,\ldots,n),$$

которые могут быть построены в соответствии с определением этого сложения и в расчёте на ВС, состоящую из L машин. Последнее означает, что если алгоритм семейства $\mathcal{A}_L=\{\mathcal{A}\}$ предусматривает параллельное выполнение операций, то число их не должно превышать L.

Мы ставим себе целью охарактеризовать оптимальный алгоритм семейства $\mathcal{A}_L$, т.е. алгоритм, реализуемый за минимальное число шагов вычислений с помощью минимального числа машин. В данном случае задача решается элементарно и ответ на неё может быть сформулирован в виде следующих 2-х теорем.

ТЕОРЕМА I. Минимальное число шагов вычислений, за которое можно сложить 2 матрицы:

$$A=(a_{ik}),\quad B=(b_{ik})$$

$$(i=1,2,\ldots,m;\quad k=1,2,\ldots,n)$$

на ВС, состоящей из L машин, равно

$$h(m,n,L)=\left[\frac{mn-1}{L}\right]+1, \qquad (I)$$

где $[x]$ - целая часть числа x.

ТЕОРЕМА 2. Минимальное число машин, которое потребуется для того, чтобы сложить матрицы A и B за минимальное число $h(m,n,L)$ шагов, равно

$$N(m,n,L)=\mu z\{h(m,n,z)=h(m,n,L)\}, \qquad (2)$$

где μ - символ μ - операции.

Обе эти теоремы очевидны и не нуждаются в особом доказательстве. Напомним лишь, что по самому определению μ - операции значение функции $N(m,n,L)$ в точке (m,n,L)

равно наименьшему натуральному числу z, удовлетворяющему при данных m,n,L уравнению, стоящему в фигурных скобках [3].

Из теоремы I при $L \geqslant mn$, в частности, имеем

$$h(m,n,L)=1,$$

т.е. при $L \geqslant mn$ 2 прямоугольные матрицы размеров $m \times n$ могут быть на ВС сложены за один шаг. Минимальное число машин, которое для этого потребуется, будет равно

$$N(m,n)=\mu L\left\{\left[\frac{mn-1}{L}\right]+1=1\right\}=mn.$$

Всё это, разумеется, ясно и непосредственно.

Сделаем ещё одно замечание. Очевидно,

$$N(m,n,L)\leq L.$$

Чтобы показать, что применение μ -операции не излишне, приведем следующий

Пример. Пусть $m=n=9$, $L=20$. Тогда по формуле (I) имеем

$$h(9,9,20)=\left[\frac{81-1}{20}\right]+1=5.$$

Применяя же формулу (2), получим

$$N(9,9,20)=\mu L\left\{\left[\frac{81-1}{L}\right]+1=5\right\}=17.$$

Нетрудно проверить, что меньшим числом машин сложить 2 квадратные матрицы порядка n = 9 за 5 шагов нельзя.

§ 2. Умножение 2-х прямоугольных матриц. Умножение прямоугольной матрицы на вектор. Скалярное произведение 2-х векторов

2.I. Умножение 2-х прямоугольных матриц. По определению произведением 2-х прямоугольных матриц:

$$A=(a_{ik}),\quad B=(b_{kj})$$

$$(i=1,2,\ldots,m;\quad k=1,2,\ldots,n;\quad j=1,2,\ldots,q)$$

называется матрица $c=(c_{ij})$, элементы которой вычисляются по формулам:

$$c_{ij}=\sum_{k=1}^{n} a_{ik}b_{kj}$$

$$(i=1,2,\ldots,m,\ j=1,2,\ldots,q).$$

Если мы имеем ВС, состоящую из L машин, то в соответствии с данным определением мы можем строить также и такие алгоритмы вычисления матрицы $C=AB$, которые предусматривают параллельное выполнение тех или иных операций. Число параллельно выполняемых операций на каждом шаге вычислений не должно, конечно, превышать L .

Пусть $\mathcal{A}_L$ - семейство всех таких алгоритмов вычисления произведения $C=AB$.

Оптимальный алгоритм семейства $\mathcal{A}_L$ будет характеризоваться некоторой длиной $h(m,n,q,L)$ и некоторым числом машин $N(m,n,q,L)$ [I]. Основная наша цель, как и ранее, состоит в том, чтобы найти выражения для указанных функций. Для этого нам понадобятся символы некоторых примитивно рекурсивных функций. Именно, функции:

$$\text{I. } sgx=\begin{cases}1, \text{ если } x>0;\\ 0, \text{ если } x=0;\end{cases}$$

$$\text{2. } \overline{sg}x=\begin{cases}0, \text{ если } x>0;\\ 1, \text{ если } x=0;\end{cases}$$

$$\text{3. } x\dot{-}y=\begin{cases}x-y, \text{ если } x\geq y;\\ 0, \text{ если } x<y.\end{cases}$$

Заметим ещё, что общее число операций сложения и умножения, необходимое для получения $C=AB$, будет равно

$$l=mq(2n-1).$$

Характеристику оптимального алгоритма семейства $\mathcal{A}_L$ по длине выражает следующая

ТЕОРЕМА 3. Минимальное число шагов, за которое можно вычислить матрицу $C=AB$ на ВС, состоящей из L машин, равно

$$h(m,n,q,L)=\begin{cases}\left[\log(n-1)\right]+2, & L\geq mnq;\\ \left[\frac{\alpha\dot{-}1}{L}\right]+2+\rho-\overline{sg}\alpha-sg(mq\dot{-}L), & 1\leq L\leq mnq,\end{cases} \tag{3}$$

где

$$\alpha=mq(2n-1)-\overline{sg}(mq\dot{-}L)\sum_{i=0}^{\rho}mq2^i,$$

$$\rho=\left[\log\frac{L}{mq}\right]\cdot\overline{sg}(mq\dot{-}L),$$

а символ $[\log x]$ означает целую часть двоичного логарифма числа x.

ДОКАЗАТЕЛЬСТВО. Пусть $\mathcal{A}$ обозначает семейство всевозможных алгоритмов умножения матрицы A на матрицу B, которые могут быть построены в соответствии с определением этого умножения и при условии, что никаких ограничений на число параллельно выполняемых операций со стороны числа машин в ВС не наложено. Очевидно, $\mathcal{A}_L\subseteq\mathcal{A}$.

Вопрос об оптимальном алгоритме семейства $\mathcal{A}$ для случая квадратных матриц A и B был изучен в работе [2]. Эти результаты легко обобщаются на случай умножения прямоугольных матриц:

$$A=(a_{ik}),\quad B=(b_{kj}),$$

$$(i=1,2,\ldots,m;\ k=1,2,\ldots,n;\ j=1,2,\ldots,q).$$

Мы получаем при этом, что оптимальный алгоритм семейства $\mathcal{A}$ характеризуется длиной

$$h=\left[\log(n-1)\right]+2=\rho+2 \tag{4}$$

и реализуется числом машин, равным

$$N=\begin{cases}mq(n-2^{\rho-1}), & \text{если } 2^{\rho}<n<2^{\rho}+2^{\rho-1},\\ 2mq(n-2^{\rho}), & \text{если } 2^{\rho}+2^{\rho-1}\leq n\leq 2^{\rho+1}.\end{cases} \tag{5}$$

Таким образом, здесь, как и всюду дальше, предполагается, что n лежит в пределах $2^{\rho}<n\leq 2^{\rho+1}$, $(\rho=0,1,2,\ldots)$.

Так как, очевидно,

$$mqn\geq N,$$

то при $L\geq mqn$ мы имеем $\mathcal{A}_L\equiv\mathcal{A}$, и потому, ссылаясь на формулу (4), мы можем утверждать, что для указанных L теорема 3 верна.

Пусть теперь

$$1\leq L<mqn.$$

Доказательство теоремы для этого случая будет состоять из построения некоторого алгоритма семейства $\mathcal{A}_L$ минимальной длины и оценки последней.

Предварительно построим один алгоритм семейства $\mathcal{A}$ минимальной длины

$$h=\left[\log(n-1)\right]+2=\rho+2 .$$

Получение матрицы $C=AB$ сводится, очевидно, к вычислению mq скалярных произведений

$$c_{ij}=\sum_{\kappa=1}^{n} a_{i\kappa}\, b_{\kappa j} \qquad (6)$$

$$(i=1,2,\ldots,m\ ;\ \ j=1,2,\ldots,q)\ .$$

Алгоритм минимальной длины $\rho+2$ вычисления c_{ij} $(i=1,2,\ldots,m;\ j=1,2,\ldots q)$ с последовательностью групповых операций $S_\kappa^{(i,j)}$ $(\kappa=1,2,\ldots,\rho+2)$ мы определим следующим образом.

1) Групповую операцию $S_1^{(i,j)}$ определим как совокупность любых $2(n-2^\rho)$ операций умножения, связанных с формулой (6). Таким образом,

$$\left|S_1^{(i,j)}\right|=2(n-2^\rho)\ .$$

2) Групповую операцию $S_2^{(i,j)}$ определим как совокупность $n-2^\rho$ операций сложения, произведенных над результатами операций, выполненных на 1-м шаге вычислений, и $2^{\rho+1}-n$ оставшихся операций умножения. Очевидно,

$$\left|S_2^{(i,j)}\right|=2^\rho .$$

3) Групповую операцию $S_\tau^{(i,j)}$ $(\tau=3,4,\ldots,\rho+2)$ определяем как совокупность $2^{\rho-\tau+2}$ операций сложения, произведённых попарно над результатами операций группы $S_{\tau-1}^{(i,j)}$. Таким образом,

$$\left|S_\tau^{(i,j)}\right|=2^{\rho-\tau+2} \qquad (\tau=3,4,\ldots,\rho+2)\ .$$

Строим теперь алгоритм с последовательностью

$$S_1, S_2, \ldots, S_{\rho+2}$$

групповых операций, определенных равенствами :

$$S_\kappa=\bigcup_{i=1}^{m}\bigcup_{j=1}^{q} S_\kappa^{(i,j)} \qquad (\kappa=1,2,\ldots,\rho+2)\ .$$

Это будет, очевидно, некоторый алгоритм вычисления $C=AB$, принадлежащий семейству $\mathcal{A}$ и имеющий минимальную длину $\rho+2$. Мощности групповых операций S_κ этого алгоритма будут удовлетворять соотношениям :

$$|S_1|=2mq(n-2^\rho)\ ,\ \ |S_\kappa|=mq\cdot 2^{\rho-\kappa+2}\ ,\ \ \kappa=2,3,\ldots,\rho+2\ .$$

Будем теперь различать 2 случая:

а) $1\le L<mq$,

б) $mq\le L<n\cdot mq$.

а) Пусть действует первое из этих предположений относительно L. В этом случае, исходя из построенного выше алгоритма вычисления $C=AB$, индукцией по κ легко показать, что операции, входящие в S_κ $(\kappa=1,2,\ldots,\rho+2)$, можно перераспределить по новым группам $\bar S_1, \bar S_2, \ldots, \bar S_{\bar h}$ так, что мощность каждой группы, кроме, быть может, последней $\bar S_{\bar h}$, будет равна L. В самом деле, для $\kappa=1$ это утверждение верно, ибо в S_1 входят лишь операции умножения. Пусть уже операции, входящие в $S_1,\ldots,S_\kappa$, могут быть распределены по новым группам :

$$\bar S_1, \ldots, \bar S_{\bar h_\kappa}$$

так, что

$$|\bar S_1|=\ldots=\left|\bar S_{\bar h_{\kappa-1}}\right|=L\ ,\quad \left|\bar S_{\bar h_\kappa}\right|\le L\ .$$

Покажем, что то же самое будет верно относительно операций, входящих в $S_1, S_2, \ldots, S_{\kappa+1}$. Для этого нам надо показать, что при $\left|\bar S_{\bar h_\kappa}\right|<L$ групповая операция $\bar S_{\bar h_\kappa}$ может быть пополнена до мощности L за счет соответствующих операций из $S_{\kappa+1}$. Так как $S_{\kappa+1}>L$, то указанное пополнение формально осуществимо. Верное неравенство

$$\left|S_{\kappa+1}\right|-\left(L-\left|\bar S_{\bar h_\kappa}\right|\right)>\left|\bar S_{\bar h_\kappa}\right|$$

гарантирует нам то, что указанное пополнение $\bar S_{\bar h_\kappa}$ до мощности L может быть осуществлено и фактически: все включенные в $\bar S_{\bar h_\kappa}$ операции будут выполнимы на $\bar h_\kappa$ шаге вычислений. Доказательство по индукции закончено.

Алгоритм с последовательностью групповых операций $\bar S_\kappa$ $(\kappa=1,2,\ldots,\bar h)$ будет принадлежать соответствующему семейству $\mathcal{A}_L$, а длина его, как нетрудно понять, выразится формулой

$$\bar h=\left[\frac{\ell\div 1}{L}\right]+1=\left[\frac{mq(2n-1)\div 1}{L}\right]+1\ . \qquad (7)$$

Очевидно также, что полученная формула выражает минимальную длину алгоритмов семейства $\mathcal{A}_L$ при $1\le L<mq$.

Чтобы окончательно убедиться в том, что при $1 \le L < mq$ теорема 3 также верна, заметим, что при указанных L формула (7) тождественна с формулой (3). Это следует из равенств:

$$\overline{sg}(mq \dot{-} L) = 0 \; ; \quad \rho = 0 \; ; \quad sg(mq \dot{-} L) = 1 \; ;$$

$$\alpha = mq(2n-1) = l \; ; \quad \overline{sg}\alpha = 0 \; .$$

б) Пусть теперь

$$mq \le L < mqn \qquad (2^{\rho} < n \le 2^{\rho+1}) \; .$$

Тогда можно указать такое $\kappa = 0, 1, 2, \ldots, \rho$, что

$$mq2^{\kappa} \le L < mq2^{\kappa+1} \; .$$

Отсюда

$$\kappa \le \log \frac{L}{mq} < \kappa + 1 \; ,$$

т.е.

$$\kappa = \left[\log \frac{L}{mq}\right] \; .$$

В данном случае операции, входящие в группы

$$S_1, S_2, \ldots, S_{\rho-\kappa+1} \; , \qquad \kappa = 0, 1, 2, \ldots, \rho \; ,$$

допускают такое перераспределение по новым группам

$$\bar{S}_1, \bar{S}_2, \ldots, \bar{S}_{h_{\rho-\kappa+1}} \; ,$$

где

$$h_{\rho-\kappa+1} = \left[\frac{\sum_{i=1}^{\rho-\kappa+1} |S_i| - 1}{L}\right] + 1 = \left[\frac{(l - \sum_{i=0}^{\kappa} mq2^i) - 1}{L}\right] + 1 \; , \quad (8)$$

что мощность каждой группы $\bar{S}_i$, кроме, быть может, последней $\bar{S}_{h_{\rho-\kappa+1}}$, будет равна L, т.е.

$$|\bar{S}_1| = \ldots = |\bar{S}_{h_{\rho-\kappa+1}-1}| = L \; ; \quad |\bar{S}_{h_{\rho-\kappa+1}}| \le L \; .$$

Это утверждение легко может быть доказано по аналогии с индуктивным доказательством, проведенным в п. а), поэтому мы на этом не останавливаемся.

Строим теперь алгоритм семейства $\mathcal{A}_L$ с последовательностью групповых операций

$$\bar{S}_1, \bar{S}_2, \ldots, \bar{S}_{h_{\rho-\kappa+1}}, \bar{S}_{1+h_{\rho-\kappa+1}}, \ldots, \bar{S}_{\bar{h}} \; ,$$

в которой первые $h_{\rho-\kappa+1}$ групповых операций определяются указанным выше перераспределением операций, входящих в группы S_i $(i = 1, 2, \ldots, \rho-\kappa+1)$, а последующие определены равенствами

$$\bar{S}_{1+h_{\rho-\kappa+1}} = S_{\rho-\kappa+2} \; , \quad \bar{S}_{2+h_{\rho-\kappa+1}} = S_{\rho-\kappa+3} \; , \ldots, \bar{S}_{\bar{h}} = S_{\rho+2} \; .$$

Легко видеть, что длина построенного алгоритма будет равна

$$\bar{h} = h_{\rho-\kappa+1} + \kappa + 1 \; . \quad (9)$$

Формула (9) при $mq \le L < mqn$, $2^{\rho} < n \le 2^{\rho+1}$, выражает наименьшую из длин алгоритмов соответствующего семейства $\mathcal{A}_L$. В самом деле, пусть

$$\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_t$$

есть последовательность групповых операций, связанная с произвольным алгоритмом семейства $\mathcal{A}_L$. По-прежнему, считая, что

$$mq2^{\kappa} \le L < mq2^{\kappa+1}, \quad \kappa = 0, 1, \ldots, \rho \; ,$$

имеем

$$t \ge \rho + 2 > \kappa + 1 \; .$$

Кроме того, справедливо будет неравенство

$$|\mathcal{G}_i| \le mq2^{t-i}, \quad i = 1, 2, \ldots, t \; ,$$

которое может быть доказано индуктивно точно так же, как доказывалось аналогичное неравенство в работе [I]. Следовательно,

$$t - (\kappa+1) \ge \left[\frac{(l - \sum_{i=0}^{\kappa} mq2^i) \dot{-} 1}{L}\right] + 1 = \left[\frac{\sum_{i=1}^{\rho-\kappa+1} |S_i| \dot{-} 1}{L}\right] + 1 = h_{\rho-\kappa+1} \; ,$$

т.е.

$$t \ge h_{\rho-\kappa+1} + \kappa + 1 = \bar{h} \; .$$

Нам остается только показать, что формула (9) для соответствующих L тождественна с формулой (3). Подставляя в (9) значения для κ и $h_{\rho-\kappa+1}$, получаем

$$\bar{h} = \left[\frac{(mq(2n-1) - \sum_{i=0}^{[\log \frac{l}{mq}]} mq2^i) - 1}{L}\right] + 2 + \left[\log \frac{L}{mq}\right] \; . \quad (10)$$

Так как при

$$mq \le L < mqn \; , \quad 2^{\rho} < n \le 2^{\rho+1} \; ,$$

справедливо

$$\overline{sg}(mq \div L) = 1 \ , \quad sg(mq \div L) = 0 \ ,$$

$$\alpha > 0 \ , \qquad \overline{sg}\alpha = 0 \ ,$$

то формула (IO) действительно будет тождественна с формулой (3). Таким образом, для рассмотренного случая б) теорема 3 также справедлива.

Покажем, наконец, что нижняя часть формулы (3) сохраняет силу и при

$$L = mqn \qquad (2^p < n \le 2^{p+1}) \ .$$

Действительно, если $L = mqn$, то

$$h(m,n,q,L) = [\log(n-1)] + 2 = p + 2 \ .$$

Применяя же формулу (3), мы имеем

$$p = \left[\log_{\frac{L}{mq}}\right] \overline{sg}(mq \div L) = \begin{cases} p+1, & L = mqn, \quad n = 2^{p+1}; \\ p, & L = mqn, \quad 2^p < n < 2^{p+1}; \end{cases}$$

$$\alpha = l - \overline{sg}(mq \div L) \sum_{i=0}^{p} mq 2^i = \begin{cases} 0, & n = 2^{p+1}; \\ 2(n-2^p)mq, & 2^p < n < 2^{p+1}. \end{cases}$$

Далее, очевидно, что

$$2(n-2^p)mq < mqn, \quad (2^p < n < 2^{p+1}),$$

и потому для $L = mqn$ и любого n, $2^p < n \le 2^{p+1}$ имеет место равенство

$$\left[\frac{\alpha \div 1}{L}\right] = 0 \ .$$

Наконец,

$$\overline{sg}\alpha = \begin{cases} 1, & L = mqn \ , \quad n = 2^{p+1}; \\ 0, & L = mqn \ , \quad 2^p < n < 2^{p+1} \ . \end{cases}$$

Следовательно, по формуле (3) при $L = mqn$ мы имеем

$$h(m,n,q,L) = \left[\frac{\alpha \div 1}{L}\right] + 2 + p - \overline{sg}\alpha - sg(mq \div L) = p + 2 \ ,$$

т.е. то же самое.

Теорема 3 полностью доказана.

Переходим к оценке оптимального алгоритма семейства $\mathcal{U}_L$ по числу машин.

Как показывают примеры, оптимальный алгоритм семейства $\mathcal{U}_L$ не всегда характеризуется L машинами. Точную характеристику оптимального алгоритма $\mathcal{A} \in \mathcal{U}_L$ по числу машин даёт следующая

ТЕОРЕМА 4. Минимальное число машин, которое потребуется для вычисления матрицы $C = AB$ на ВС, состоящей из L машин, за $h(m,n,q,L)$ шагов, равно

$$N(m,n,q,L) = \mu z \left\{ h(m,n,q,z) = h(m,n,q,L) \right\}, \quad \text{(II)}$$

где μ - символ μ - операции.

Эта теорема, очевидно, не требует особого доказательства. Заметим только, что в формуле (II) для выражения функции $h(m,n,q,L)$ можно всегда пользоваться нижней частью формулы (3).

В частности, полагая

$$h(m,n,q,L) = [\log(n-1)] + 2 \ ,$$

мы получим характеристику по числу машин для оптимального алгоритма семейства $\mathcal{U}$:

$$N = N(m,n,q) = \mu L \left\{ \left[\frac{\alpha \div 1}{L}\right] + 2 + p - \overline{sg}\alpha - sg(mq \div L) = [\log(n-1)] + 2 \right\},$$

где

$$\alpha = mq(2n-1) - \overline{sg}(mq \div L) \sum_{i=0}^{p} mq 2^i \ ,$$

$$p = \left[\log_{\frac{L}{mq}}\right] \overline{sg}(mq \div L) \ .$$

Явное выражение этой корневой функции $N(m,n,q)$ дают формулы (5).

Отметим теперь частные случаи доказанной теоремы 3, относящиеся к умножению матрицы на вектор и вычислению скалярного произведения 2-х векторов.

2.2. Умножение прямоугольной матрицы на вектор. Характеристику оптимального алгоритма умножения прямоугольной матрицы

$$A = (a_{ik})$$

$$(i = 1,2,\ldots m \ ; \quad k = 1,2,\ldots n)$$

на вектор $b=(b_1, b_2, \dots b_n)$ мы получим, полагая в предыдущих рассуждениях $q=1$. В результате мы имеем следующие 2 теоремы.

ТЕОРЕМА 5. Минимальное число шагов, за которое можно вычислить произведение матрицы $A=(a_{ik})$ на вектор $b=(b_i)_1^n$ на ВС, состоящей из L машин, равно

$$h(m,n,L)=\begin{cases}\left[\log(n-1)\right]+2, & L\geqslant mn,\\ \left[\frac{\alpha\div 1}{L}\right]+2+\rho-\overline{sg}\,\alpha-sg(mq\div L), & 1\leqslant L\leqslant mn,\end{cases}$$

где

$$\alpha=m(2n-1)-\overline{sg}(m\div L)\sum_{i=0}^{\rho} m2^i,$$

$$\rho=\left[\log\frac{L}{m}\right]\overline{sg}(m\div L).$$

ТЕОРЕМА 6. Минимальное число машин, которое потребуется для того, чтобы вычислить $c=Ab$ за $h(m,n,L)$ шагов равно

$$N(m,n,L)=\mu z\left\{h(m,n,z)=h(m,n,L)\right\}.$$

В одном важном частном случае, когда $h(m,n,L)=\left[\log(n-1)\right]+2$, функция $N(m,n,L)$ может быть записана явно в виде формул (5) при $q=1$.

2.3. Скалярное произведение 2-х векторов. Оптимальный алгоритм вычисления на ВС скалярного произведения 2-х векторов $a=(a_i)_1^n$ и $b=(b_i)_1^n$, характеризуется следующими двумя теоремами, являющимися следствием теорем 3 и 4 при $m=q=1$.

ТЕОРЕМА 7. Минимальное число шагов, за которое можно вычислить скалярное произведение 2-х векторов

$$(a,b)=\sum_{i=1}^{n} a_i b_i$$

на ВС, состоящей из L машин, равно

$$h(n,L)=\begin{cases}\left[\log(n-1)\right]+2, & L\geqslant n,\\ \left[\log 4L\right]+\left[\frac{\alpha\div 1}{L}\right]-\overline{sg}\,\alpha, & 1\leqslant L\leqslant n,\end{cases}$$

где

$$\alpha=2n-1-\sum_{i=0}^{[\log L]} 2^i.$$

ТЕОРЕМА 8. Чтобы вычислить на ВС скалярное произведение 2-х векторов $a=(a_i)_1^n$ и $b=(b_i)_1^n$, за $h(n,L)$ шагов, необходимо как минимум

$$N(n)=\mu z\left\{h(n,z)=h(n,L)\right\}$$

машин.

В частности, как это следует из формул (5) при $m=q=1$

$$N(n)=\mu L\left\{\left[\log 4L\right]+\left[\frac{\alpha\div 1}{L}\right]-\overline{sg}\,\alpha=\left[\log(n-1)\right]+2\right\}_{[\log L]}=\begin{cases}n-2^{\rho-1}, & 2^{\rho}<n<2^{\rho}+2^{\rho-1},\\ 2(n-2^{\rho}), & 2^{\rho}+2^{\rho-1}\leqslant n\leqslant 2^{\rho+1},\end{cases}$$

$$\left(\rho=\left[\log(n-1)\right],\ \alpha=2n-1-\sum_{i=0}^{[\log L]}2^i\right).$$

Этот результат был указан в работе [2].

Заканчивая рассмотрение вычисления на ВС скалярного произведения 2-х векторов, сделаем ещё одно

Замечание. В случае скалярного произведения для функции $h(n,L)$ имеет место также следующая формула:

$$h(n,L)=\left[\frac{2n}{L}\right]+\left[\log\left(2n-L\left[\frac{2n}{L}\right]+L-1\right)\right]. \qquad (12)$$

Для доказательства этой формулы построим следующий алгоритм вычисления (a,b), принадлежащий соответствующему семейству $\mathcal{A}_L$. Сначала выполняем операции умножения по L операций на каждом шаге. Число таких шагов будет равно $\left[\frac{n}{L}\right]$. На $\left[\frac{n}{L}\right]+1$ шаге выполним

$$n-L\left[\frac{n}{L}\right]$$

оставшихся операций умножения и

$$L-\left(n-L\left[\frac{n}{L}\right]\right)=L-n+L\left[\frac{n}{L}\right]$$

допустимых операций сложения. Если предположить, что

$$2 \le L \le \left[\frac{n}{2}\right] ,$$

то мы имеем

$$\left[\frac{n}{L}\right] \ge 2 .$$

Отсюда, как легко видеть, указанный процесс вычислений действительно может быть организован. После этого нам нужно будет сложить еще

$$m = n - \left(L - n - L\left[\frac{n}{L}\right]\right) = 2n - L - L\left[\frac{n}{L}\right]$$

чисел. Очевидно,

$$m \ge \left[\frac{n}{2}\right] \ge L$$

и потому, как показано в работе [I], минимальное число шагов, за которое можно сложить n чисел с помощью L машин равно

$$h_1 = \left[\frac{n}{L}\right] + \left[\log\left(n - L\left[\frac{n}{L}\right] + L - 1\right)\right] .$$

Общее число шагов вычислений выразится тогда формулой

$$h = h_1 + \left[\frac{n}{L}\right] + 1 .$$

Заметив, что

$$\left[\frac{n}{L}\right] = \left[\frac{2n}{L}\right] - 1 - \left[\frac{n}{L}\right] ,$$

и сделав соответствующую подстановку значений для n и $\left[\frac{n}{L}\right]$, мы получим формулу (I2). В процессе доказательства формулы (I2) мы предполагали, что L лежит в пределах

$$2 \le L \le \left[\frac{n}{2}\right] .$$

Нетрудно, однако, показать, что формулой (I2) можно пользоваться и при

$$2 \le L \le n .$$

§ 3. Умножение треугольной матрицы на прямоугольную. Умножение треугольной матрицы на вектор

Часто приходится иметь дело с умножением треугольной матрицы на вектор или на другую матрицу. Поэтому рассмотрение указанных задач под прежним углом зрения представляет самостоятельный интерес. В то же время результаты § 2 не могут быть

сюда перенесены непосредственно, и мы вынуждены этот вопрос рассмотреть особо.

3.I. Умножение треугольной матрицы на прямоугольную.

Пусть

$$T = U \cdot V ,$$

где

$$U = (u_{ik})_1^n , \qquad u_{ik} = 0 , \quad k > i ,$$

$$V = (v_{kj}) , \qquad k = 1, 2, \ldots n , \quad j = 1, 2, \ldots q .$$

Для определенности мы взяли умножение слева нижней треугольной матрицы U на прямоугольную матрицу V .Однако наши результаты не изменятся, если мы будем рассматривать умножение справа, или вместо U возьмем верхнюю треугольную матрицу.

Пусть $\mathcal{A}$ - семейство всевозможных алгоритмов вычисления $T = U \cdot V$, которые могут быть вообще построены в соответствии с определением умножения матриц (с учётом треугольности матрицы U), а $\mathcal{A}_L$ - подсемейство семейства $\mathcal{A}$ тех алгоритмов вычисления T , которые могут быть построены при одном дополнительном условии, что на каждом шаге вычислений допустимо параллельно выполнять не более L операций сложения и умножения. Наша задача состоит в оценке оптимальных алгоритмов семейств $\mathcal{A}$ и $\mathcal{A}_L$.

Характеристика оптимального алгоритма семейства $\mathcal{A}$ по длине дана, по существу, в работе [2] . Так же, как и для случая умножения 2-х квадратных матриц, наименьшая из длин алгоритмов семейства $\mathcal{A}$ будет равна

$$h = \left[\log(n-1)\right] + 2 = \rho + 2 ,$$

считая, что

$$2^\rho < n \le 2^{\rho+1} , \qquad \rho = 0, 1, 2, 3, \ldots .$$

Характеристику оптимального алгоритма семейства $\mathcal{A}$ по числу машин получим после того, как найдем характеристику $h(n, q, L)$ оптимального алгоритма семейства $\mathcal{A}_L$ по длине. Для этой цели нам понадобится один алгоритм семейства $\mathcal{A}$ минимальной длины $\rho + 2$, который мы опишем следующим образом.

Вычисление матрицы $T = UV$ сводится, очевидно, к получению nq скалярных произведений вида

$$t_{ij}=\sum_{\kappa=1}^{i} u_{i\kappa}\,v_{\kappa j}\ ,$$

$$(i=1,2,\dots n;\quad j=1,2,\dots q)\ .$$

Алгоритм вычисления t_{ij} $(i\geqslant 2)$ минимальной длины $h_i=\lceil\log(i-1)\rceil+2$ мы определим с помощью такой последовательности

$$S_1^{(i,j)},\ S_2^{(i,j)},\ \dots,\ S_{h_i}^{(i,j)} \qquad (h_n=\rho+2)$$

групповых операций.

1) I-й шаг вычислений, т.е. групповую операцию $S_1^{(i,j)}$, определим как совокупность $2(i-2^{\lceil\log(i-1)\rceil})$ операций умножения компонент векторов

$$u_i=(u_{i1},u_{i2},\dots,u_{ii}),\quad v_j=(v_{1j},v_{2j},\dots,v_{ij})\ .$$

2) Групповую операцию $S_2^{(i,j)}$ определим как совокупность $i-2^{\lceil\log(i-1)\rceil}$ операций сложения, произведенных попарно над результатами операций из $S_1^{(i,j)}$, и

$$2^{\lceil\log(i-1)\rceil+1}-i\ ,$$

оставшихся из числа i операций умножения компонент векторов u_i и v_j.

3) Каждая последующая групповая операция $S_\kappa^{(i,j)}$ $(\kappa=3,4,\dots h_i)$ состоит из операций сложения, произведенных попарно над результатами операций предыдущей группы $S_{\kappa-1}^{(i,j)}$.

Очевидно, мощности групповых операций $S_\kappa^{(i,j)}$ удовлетворяют следующим соотношениям

$$|S_1^{(i,j)}|=2(i-2^{\lceil\log(i-1)\rceil}),\quad |S_\kappa^{(i,j)}|=2^{\lceil\log(i-1)\rceil-\kappa+2}\ , \qquad (I3)$$

$$i=2,3,\dots n;\quad j=1,2,\dots,q$$

$$\kappa=2,3,\dots,h_i;\quad h_i=\lceil\log(i-1)\rceil+2\ .$$

При $i=1$ вычисление t_{ij} сводится к вычислению произведения $u_{11}v_{1j}$, следовательно,

$$|S_1^{(1,j)}|=1;\quad h_1=1\ , \qquad (I3a)$$

$$(j=1,2,\dots,q)\ .$$

Нужный нам алгоритм вычисления $T=UV$ минимальной длины $\rho+2$ строится теперь с помощью последовательности групповых операций

$$S_1,\ S_2,\ \dots\dots,\ S_{\rho+2}\ ,$$

определенных равенствами:

$$S_1=\bigcup_{j=1}^{q}\bigcup_{i=1}^{n} S_{h_i-(\rho+1)}^{(i,j)}\ ,$$

$$S_2=\bigcup_{j=1}^{q}\bigcup_{i=1}^{n} S_{h_i-\rho}^{(i,j)}\ ,$$

$$\dots\dots\dots\dots\dots \qquad (I4)$$

$$S_\kappa=\bigcup_{j=1}^{q}\bigcup_{i=1}^{n} S_{h_i-(\rho-\kappa+2)}^{(i,j)}\ ,$$

$$\dots\dots\dots\dots\dots$$

$$S_{\rho+2}=\bigcup_{j=1}^{q}\bigcup_{i=1}^{n} S_{h_i}^{(i,j)}\ ,$$

где считается, что при

$$h_i-(\rho-\kappa+2)\leqslant 0$$

соответствующая групповая операция - пустая:

$$S_{h_i-(\rho-\kappa+2)}^{(i,j)}=\phi\ .$$

Пользуясь формулами (I3), (I3а) и (I4) нетрудно подсчитать, что мощности групповых операций S_κ удовлетворяют соотношениям:

$$l_1=|S_1|=q\sum_{i=2^\rho}^{n}2(i-2^\rho)\ ,$$

$$l_2=|S_2|=q\left\{2^\rho(n-2^\rho+1)+\sum_{\kappa=1}^{2^{\rho-1}}2\kappa\right\}\ ,$$

$$\dots\dots\dots\dots\dots$$

$$l_j=|S_j|=q\left\{2^{\rho-j+2}(n-2^{\rho-j+2}+1)+\sum_{\kappa=1}^{2^{\rho-j+1}-1}2\kappa\right\}\ , \qquad (I5)$$

$$\dots\dots\dots\dots\dots$$

$$l_{\rho+1}=|S_{\rho+1}|=q\left\{2^1(n-2^1+1)+\sum_{\kappa=1}^{\kappa=0}2\kappa\right\}=2q(n-1)\ ,$$

$$l_{\rho+2}=|S_{\rho+2}|=q\left\{2^0(n-2^0+1)\right\}=qn\ ,$$

$$(2^\rho<n\leqslant 2^{\rho+1})\ .$$

Отметим теперь ряд соотношений между l_i, которые нам понадобятся в дальнейшем.

I. Прежде всего заметим, что общее число арифметических операций "+" и "×", необходимых для получения матрицы $T=U\cdot V$, равно

$$l=q\sum_{i=1}^{n}(2i-1)=q\left\{2\cdot\frac{(1+n)n}{2}-n\right\}=qn^2. \tag{I6}$$

Следовательно,

$$l=\sum_{i=1}^{\rho+2}l_i=qn^2 \quad (2^\rho<n\le 2^{\rho+1}). \tag{I7}$$

2. Справедливо неравенство

$$l_1\le l_2. \tag{I8}$$

В самом деле, считая по-прежнему, что

$$2^\rho<n\le 2^{\rho+1} \quad (\rho=0,1,2,\dots),$$

представим n в виде

$$n=2^\rho+j, \quad j=1,2,\dots,2^\rho.$$

По формуле (I5) имеем

$$l_1=q\sum_{i=2^\rho}^{n}2(i-2^\rho)=2q\sum_{\kappa=1}^{n-2^\rho}\kappa=q(n-2^\rho)(n-2^\rho+1)=q(j^2+j),$$

$$l_2=q\left\{2^\rho(n-2^\rho+1)+\sum_{\kappa=1}^{2^{\rho-1}-1}2\kappa\right\}=q\left\{2^\rho(n-2^\rho+1)+2^{\rho-1}(2^{\rho-1}-1)\right\}=q(2j^\rho+2^\rho+2^{\rho-1}(2^{\rho-1}-1)).$$

Так как

$$j\le 2^\rho,$$

то ясно, что $l_1\le l_2$, причём равенство возможно лишь при $\rho=1,0$ и $j=2^\rho$, т.е. при $n=4$ и при $n=2$ (в последнем случае l_2 вычисляем, как $l_{\rho+2}$ при $\rho=0$).

3. При любом

$$2^\rho<n\le 2^{\rho+1} \quad (\rho=1,2,3,\dots),$$

будет

$$l_j>l_{j+1} \quad (j\ge 3) \tag{I9}$$

Действительно, по формуле (I5) имеем

$$l_j=q\left\{2^{\rho-j+2}(n-2^{\rho-j+2}+1)+\sum_{\kappa=1}^{2^{\rho-j+1}-1}2\kappa\right\}=q\left\{2^{\rho-j+2}(n-2^{\rho-j+2}+1)+2^{\rho-j+1}(2^{\rho-j+1}-1)\right\},$$

$$l_{j+1}=q\left\{2^{\rho-j+1}(n-2^{\rho-j+1}+1)+\sum_{\kappa=1}^{2^{\rho-j}-1}2\kappa\right\}=q\left\{2^{\rho-j+1}(n-2^{\rho-j+1}+1)+2^{\rho-j}(2^{\rho-j}-1)\right\}.$$

Заменяя в этих выражениях n согласно равенству

$$n=2^\rho+i, \quad i=1,2,\dots,2^\rho,$$

и делая несложные преобразования, получим

$$l_j=2^{\rho-j}q\left\{2^{\rho+1}+4i+2+2^{\rho-j+2}+(2^{\rho+1}-2^{\rho-j+4})\right\}, \tag{20}$$

$$l_{j+1}=2^{\rho-j}q\left\{2^{\rho+1}+2i+1+2^{\rho-j}-2^{\rho-j+2}\right\}.$$

Очевидно, при $j\ge 3$ мы имеем

$$2^{\rho+1}-2^{\rho-j+4}\ge 0$$

и потому

$$l_j>l_{j+1} \quad (j=3,4,\dots,\rho+1).$$

4. При $\rho\ge 3$ справедливы неравенства:

$$\begin{aligned} &l_2>l_3, \text{ если } && 2^\rho+2^{\rho-3}\le n<2^{\rho+1},\\ &l_2<l_3, \text{ если } && 2^\rho<n<2^\rho+2^{\rho-3}. \end{aligned} \tag{2I}$$

Из формул (20) при $j=2$ имеем

$$l_2=2^{\rho-2}q\left\{4i+2+2^\rho\right\},$$

$$l_3=2^{\rho-2}q\left\{2^\rho+2i+1+2^{\rho-2}\right\}.$$

Отсюда следует, что неравенство $l_2\ge l_3$ возможно лишь при

$$4i+2+2^\rho\ge 2i+1+2^{\rho-2}+2^\rho,$$

т.е. при

$$i\ge 2^{\rho-3}-\frac{1}{2}.$$

Так как i должно быть целым, то при $i\ge 2^{\rho-3}$ мы имеем

$$l_2>l_3.$$

Очевидно, при $2\le n\le 2^4$ всегда

$$l_2>l_3.$$

5. Всегда

$$l_2>l_4. \tag{22}$$

Это легко следует из сравнения выражений:

$$l_2 = 2^{\rho-2} q \left\{ 4i+2+2^{\rho} \right\} = 2^{\rho-3} q \left\{ 8i+4+2^{\rho+1} \right\},$$

$$l_4 = 2^{\rho-3} q \left\{ 2^{\rho+1} + 2i + 1 + 2^{\rho-3} - 2^{\rho-1} \right\},$$

ибо

$$8i+4 > 2i+1+2^{\rho-3}-2^{\rho-1} \quad (i=1,2,\ldots,2^{\rho}).$$

6. Имеет место равенство

$$\left[\frac{l_3}{l_4}\right] = 1. \tag{23}$$

Действительно,

$$\left[\frac{l_3}{l_4}\right] = \left[\frac{2^{\rho+1}+4i+2+2^{\rho-1}}{2^{\rho+1}+2i+1+2^{\rho-3}-2^{\rho-1}}\right] = 1+\left[\frac{2i+1-2^{\rho-3}+2^{\rho}}{2^{\rho+1}+2i+1+2^{\rho-3}-2^{\rho-1}}\right] = 1+0 = 1.$$

7. Справедливо равенство

$$\left[\frac{l_1+l_2+l_3}{l_2}\right] = 2 \quad \text{при} \quad l_3 > l_2. \tag{24}$$

В самом деле, используя написанные выше выражения для l_1, l_2, l_3 имеем

$$\left[\frac{l_1+l_2+l_3}{l_2}\right] = 1+\left[\frac{l_1+l_3}{l_2}\right] = 1+\left[\frac{i^2+i+2^{\rho-2}(2^{\rho}+2i+1+2^{\rho-2})}{2^{\rho-2}(4i+2+2^{\rho})}\right] =$$

$$= 2+\left[\frac{i^2+i+2^{\rho-2}(2^{\rho-2}-2i-1)}{2^{\rho-2}(4i+2+2^{\rho})}\right].$$

Так как по условию $l_3 > l_2$, то на основании формулы (21) будет $1 \leq i < 2^{\rho-3}$. Следовательно,

$$0 \leqslant \left[\frac{i^2+i+2^{\rho-2}(2^{\rho-2}-2i-1)}{2^{\rho-2}(4i+2+2^{\rho})}\right] \leqslant \left[\frac{i^2+i+2^{\rho-2}\cdot 2^{\rho-2}}{2^{\rho-2}(4i+2+2^{\rho})}\right] \leqslant \left[\frac{2^{\rho-3}(2^{\rho-3}+1)+2^{\rho-2}\cdot 2^{\rho-2}}{2^{\rho-2}(4i+2+2^{\rho})}\right] = 0$$

ибо

$$2^{2\rho-4}+2^{2\rho-6} < 2^{2\rho-2}.$$

Неравенство (24) тем самым доказано.

Обратимся теперь к доказательству основной теоремы настоящего параграфа. Предварительно введём новые обозначения:

$$\sigma = \max_{1 \leqslant i \leqslant \rho+2} l_i = \max(l_2, l_3),$$

$$\tau = \min_{2 \leqslant i \leqslant \rho+2} l_i = l_{\rho+2} = qn,$$

где величины l_i определены формулами (15).

ТЕОРЕМА 9. Минимальное число шагов вычислений, которое потребуется для того, чтобы на ВС, состоящей из L машин можно было выполнить умножение треугольной матрицы $U=(u_{ik})_n$ на прямоугольную матрицу $V=(v_{kj})$ $(k=1,2,\ldots n;\ j=1,2,\ldots q)$, равно

$$h(n,q,L) = \begin{cases} \left[\log(n-1)\right]+2, & L \geqslant \sigma, \\ \left[\frac{\mathcal{L}-1}{L}\right]+1+(\rho+2-\rho)+sg(L-l_2)\,sg\left\{2-\left[\frac{\mathcal{L}-1}{L}\right]\right\}, & 1 \leqslant L < \sigma, \end{cases} \tag{25}$$

где $\mathcal{L} = qn^2 - \overline{sg}(qn-L)\sum_{i=\rho+1}^{\rho+2} l_i$,

$\rho = \mu i\left\{ l_i > L \geqslant l_{i+1} \right\}$, $i = 2,3,\ldots,\rho+2$,

$\rho = \left[\log(n-1)\right]$.

ДОКАЗАТЕЛЬСТВО. Равенство

$$h(n,q,L) = \left[\log(n-1)\right]+2 = \rho+2, \quad L \geqslant \sigma$$

есть следствие предыдущих замечаний.

Пусть теперь $1 \leqslant L \leqslant \sigma$. Будем различать 2 случая:

а) $1 \leqslant L < \tau = qn$,

б) $\tau \leqslant L < \sigma$.

Пусть имеет место случай а). Так же, как в § 2, индукцией можно показать, что операции, входящие в группы

$$S_1, S_2, \ldots S_{\rho+2},$$

так распределяются по новым группам

$$\overline{S}_1, \overline{S}_2, \ldots \overline{S}_h,$$

где

$$\bar{h}=\left[\frac{L-1}{L}\right]+1=\left[\frac{qn^2-1}{L}\right]+1 ,$$

что мощность каждой группы $\bar{S}_i$, кроме, может быть, последней $\bar{S}_{\bar{h}}$, будет равна L:

$$|\bar{S}_1|=\ldots=|\bar{S}_{\bar{h}-1}|=L ; \quad |\bar{S}_{\bar{h}}|\le L .$$

Это следует из того, что в рассматриваемом случае мы имеем

$$l_i=|S_i|>L \qquad (i=2,\ldots,\rho+2).$$

Длина $\bar{h}$ получаемого при этом алгоритма соответствующего семейства $\mathcal{A}_L$ будет, очевидно, минимальной. Таким образом, при $1\le L<\tau=qn$ мы имеем

$$h(n,q,n)=\left[\frac{qn^2-1}{L}\right]+1 .$$

Однако то же самое значение для $h(n,q,L)$ мы получим и по формуле (25), ибо в данном случае

$$\overline{sg}(qn\dot{-}L)=0 , \quad \mathcal{L}=qn^2 , \quad \rho=\rho+2 ,$$

$$\left[\frac{\mathcal{L}-1}{L}\right]\ge 2 , \quad sg\left\{2\dot{-}\left[\frac{\mathcal{L}-1}{L}\right]\right\}=0 \qquad (n\ge 2).$$

Пусть теперь выполняется случай б)

$$qn=\tau\le L<\sigma=max(l_2,l_3).$$

Могут представиться 3 возможности:

I) $l_2>l_3$,

2) $l_3>l_2$, но $l_2\ge L$,

3) $l_3>l_2$ и $L>l_2$.

I) Пусть $l_2>l_3$. Можно, очевидно, указать такое $\rho=2,\ldots,\rho+1$, что

$$l_\rho>L\ge l_{\rho+1} .$$

Это ρ можно также записать в виде

$$\rho=\mu i\{l_i>L\ge l_{i+1}\} .$$

В данном случае операции, выходящие в S_i $(i=1,2,\ldots,\rho)$, могут быть так перераспределены по новым группам

$$\bar{S}_1,\bar{S}_2,\ldots\bar{S}_{h_\rho} ,$$

где

$$h_\rho=\left[\frac{\sum_{i=1}^{\rho}|S_i|-1}{L}\right]+1=\left[\frac{(qn^2-\sum_{i=\rho+1}^{\rho+2}l_i)-1}{L}\right]+1 , \qquad (26)$$

что мощности $|\bar{S}_i|$ будут удовлетворять соотношениям

$$|\bar{S}_1|=L ; \ldots . |\bar{S}_{h_{\rho-1}}|=L ; \quad |\bar{S}_{h_\rho}|\le L .$$

Это предложение может быть доказано методом полной индукции по аналогии с соответствующим доказательством, проведённым в § 2.

Алгоритм семейства $\mathcal{A}_L$ с последовательностью групповых операций:

$$\bar{S}_1,\bar{S}_2,\ldots,\bar{S}_{h_\rho},\bar{S}_{h_{\rho+1}},\ldots,\bar{S}_{\bar{h}} ,$$

где

$$\bar{S}_{h_{\rho-1}}=S_{\rho+1},\ldots,\bar{S}_{\bar{h}}=S_{\rho+2} ,$$

будет иметь длину

$$\bar{h}=h_\rho+(\rho+2-\rho). \qquad (27)$$

Так как при

$$qn=\tau\le L<\sigma , \quad l_2>l_3 ,$$

мы имеем $\overline{sg}(qn\dot{-}L)=1 ; \quad sg(L\dot{-}l_2)=0 ,$

то формула (27) после подстановки в неё выражения для h_ρ из формулы (26) может быть записана в виде

$$\bar{h}=\left[\frac{\mathcal{L}-1}{L}\right]+1+(\rho+2-\rho)+sg(L\dot{-}l_2)sg\left\{2\dot{-}\left[\frac{\mathcal{L}-1}{L}\right]\right\} . \qquad (28)$$

Формула (27) или равносильная ей формула (28) выражает наименьшую из длин соответствующего семейства $\mathcal{A}_L$.

Действительно, пусть $\mathcal{A}$ - произвольный алгоритм семейства $\mathcal{A}_L$ с последовательностью групповых операций

$$G_1,G_2,\ldots G_t .$$

На основании предыдущего ((I9), (2I)) мы имеем

$$l_2>l_3>\ldots>l_{\rho+2} ,$$

$$l_\rho>L\ge l_{\rho+1} , \quad \rho=2,\ldots,\rho+1 .$$

Очевидно, что

$$t \geq \rho+2 > \rho+2-\rho .$$

Нетрудно далее понять, что

$$\sum_{i=\rho+1}^{\rho+2} |G_i| \leq \sum_{i=\rho+1}^{\rho+2} l_i .$$

Следовательно,

$$t-(\rho+2-\rho) \geq \left[\frac{(l-\sum_{i=\rho+1}^{\rho+2}|G_i|)-1}{L}\right]+1 \geq \left[\frac{(l-\sum_{i=\rho+1}^{\rho+2} l_i)-1}{L}\right]+1 = h_\rho .$$

Отсюда

$$t \geq h_\rho+(\rho+2-\rho)=\bar{h} .$$

Таким образом,

$$h(n,q,L)=\bar{h} ,$$

и теорема 9 в этом частном случае доказана.

2) Случай $l_3 > l_2$, $l_2 \geq L$, рассматривается аналогично. Именно может быть указано такое $\rho = 3,4,\ldots,$ что

$$l_\rho > L \geq l_{\rho+1} .$$

Так же, как и в предыдущем случае, операции, входящие в S_i $(i=1,2,\ldots,\rho)$, разбиваются по новым группам $\bar{S}_j$ $(j=1,2,\ldots h_\rho)$, где h_ρ по-прежнему будет выражаться формулой (26). Далее, аналогично может быть определен соответствующий алгоритм вычисления $T=UV$ с последовательностью групповых операций $\bar{S}_i$ $(i=1,2,\ldots,\bar{h})$, где $\bar{h}$ находится из формулы (27) или из равносильной формулы (28). Наконец, как и ранее, можно показать, что $\bar{h}$ есть наименьшая из длин алгоритмов семейства $\mathcal{A}_L$, т.е. $h(n,q,L)=\bar{h}$. Таким образом, мы убедимся, что теорема 9 верна и для этого случая.

3) $l_3 > l_2$ и $L > l_2$. Кроме того, по-прежнему

$$qn=\tau \leq L < \sigma = l_3 .$$

Используя неравенство (22) и условие $L > l_2$, получаем также

$$l_3 > L > l_2 > l_4 . \qquad (29)$$

Следовательно, в данном случае

$$\rho = \mu i\{l_i > L \geq l_{i+1}\} = 3 , \qquad (30)$$

$$(i=3,\ldots,\rho+1).$$

Пусть $\bar{h}$ есть наименьшая из длин алгоритмов семейства $\mathcal{A}_L$ для указанных L. Тогда, ввиду (30), $\bar{h}=h_1+(\rho-1)$, где h_1 – число шагов, которое потребуется для выполнения операций групп S_1, S_2, S_3. Покажем, что $h_1=3$, т.е. $\bar{h}=\rho+2$. Очевидно, что

$$\bar{h} \geq \rho+2 ,$$

поэтому $h_1 \geq 3$.

Далее, в силу неравенств (23) и (29) получаем

$$\left[\frac{l_3}{L}\right]=1$$

и, значит, поскольку ещё $L > l_2 \geq l_1$, $h_1 \leq 4$.

Таким образом, $3 \leq h_1 \leq 4$.

На самом деле здесь будет равенство $h_1=3$.

Для этого достаточно показать, что операции, входящие в группы S_1, S_2, S_3 можно разбить на три группы

$$\bar{S}_1 , \bar{S}_2 , \bar{S}_3 ,$$

мощность которых не превышает L.

Пусть сначала группа S_1 мощности l_1 пополняется до мощности L за счет подходящих операций из S_2. Тогда, как нетрудно понять, остаток их может быть пополнен до группы $\bar{S}_2$ за счет операций из S_3 так, что мощность $\bar{S}_2$ будет тоже равна L: $|\bar{S}_2|=L$. Из оставшихся операций, входивших в S_3, образуем $\bar{S}_3$. Так как в данном случае в силу неравенства (24)

$$\left[\frac{|S_1|+|S_2|+|S_3|}{L}\right]=\left[\frac{l_1+l_2+l_3}{L}\right]=2 ,$$

то $h_1=3$, $|\bar{S}_3|<L$.

Если же S_1 не может быть пополнена до мощности L за счет операций из S_2, т.е.

$$|S_1|+(|S_2|-q(\sum_{i=2^\rho}^{n}(i-2^\rho))<L ,$$

то h_1 будет подавно равно 3, причём

$$|\bar{S}_1|<L , \quad |\bar{S}_2|=L , \quad |\bar{S}_3|<L .$$

Таким образом, $h_1=3$ и $\bar{h}=\rho+2$.

Нам остаётся только показать, что в рассматриваемом случае формула (25) даёт то же самое значение.

Действительно, так как

$$\rho=3, \quad sg(L-l_2)=1, \quad \overline{sg}(qn-L)=1,$$

то формула (25) принимает вид

$$h(n,q,L)=\left[\frac{L-1}{L}\right]+\rho+sg\left\{2-\left[\frac{L-1}{L}\right]\right\}, \qquad (31)$$

где

$$L=qn^2-\sum_{i=4}^{\rho+2} l_i \; ; \quad \rho=\left[\log(n-1)\right].$$

В силу неравенства (24) мы имеем

$$1\le\left[\frac{L-1}{L}\right]\le 2,$$

т.е. либо $\left[\frac{L-1}{L}\right]=1$, либо $\left[\frac{L-1}{L}\right]=2$.

Легко видеть, что в обоих случаях формула (31) даёт

$$h(n,q,L)=\rho+2.$$

Теорема 9 доказана.

Располагая функцией $h(n,q,L)$ мы можем дать характеристику оптимальных алгоритмов семейств $\mathcal{A}$ и $\mathcal{A}_L$ по числу машин.

ТЕОРЕМА IO. О п т и м а л ь н ы й а л г о р и т м с е м е й с т в а $\mathcal{A}_L$ р е а л и з у е т с я

$$N(n,q,L)=\mu z\left\{h(n,q,z)=h(n,q,L)\right\}$$

м а ш и н а м и.

ТЕОРЕМА II. О п т и м а л ь н ы й а л г о р и т м с е м е й с т в а $\mathcal{A}$ р е а л и з у е т с я

$$N(n,q)=\mu z\left\{h(n,q,z)=\left[\log(n-1)\right]+2\right\}$$

м а ш и н а м и.

В обеих теоремах выражение для $h(n,q,z)$ берётся по формуле (25).

3.2. Умножение прямоугольной матрицы на вектор. Эта задача является, очевидно, частным случаем умножения треугольной матрицы на прямоугольную матрицу. Поэтому, если в теоремах 9, IO, II положить $q=1$, то мы получим соответствующие результаты для рассматриваемой задачи.

В заключение скажем, что полученные выше оценки позволяют сделать определённые выводы относительно применения ВС к решению рассмотренных задач на основные действия с матрицами. Заметим также, что наряду с оценками оптимальных алгоритмов для решения упомянутых задач на ВС, нами построены, по существу, и сами оптимальные алгоритмы.

Л и т е р а т у р а

I. Бекишев Г.А. Об алгоритмах, эффективно реализуемых на вычислительных системах. Сб. "Вычислительные системы", вып. 7 , Новосибирск, I963.

2. Бекишев Г.А. О распараллеливании вычислительных алгоритмов. Сб. "Вычислительные системы", вып. 5, Новосибирск, I963.

3. Успенский В.А. Лекции о вычислимых функциях. М., I960.