МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ
(Вычислительные системы)
1979 год Выпуск 78

УДК 519.687.6:519.688

ПАРАЛЛЕЛЬНЫЕ ПРОГРАММЫ ДЛЯ СИСТЕМЫ МИНИМАКС

Ю.И.Колосова, И.К.Кербель, В.Г.Кербель

В многопроцессорной системе МИНИМАКС одним из важнейших режимов является режим параллельной обработки, при котором для решения одной задачи, представленной параллельной программой, выделяется подсистема из нескольких элементарных машин [I]. Под параллельной программой понимается организованная совокупность последовательных программ (ветвей), выполняющихся одновременно на отдельных элементарных машинах системы и взаимодействующих между собой в соответствии с требованиями параллельного алгоритма.

Важнейшими компонентами системы параллельного программирования в МИНИМАКСе являются языки для записи параллельных алгоритмов, получивших название ОВС-языков [2,3]. В данной работе описываются параллельные алгоритмы и программы для решения системы линейных уравнений методом последовательных приближений, обращения матрицы методом пополнения, внутренней сортировки, а также для специальной задачи, демонстрирующей организацию продолжения решения задачи при выходе из строя одной из элементарных машин подсистемы. Параллельные программы представлены на ОВС-ФОРТРАНе.

I. Решение системы линейных уравнений методом последовательных приближений

Система линейных уравнений записывается в виде $X = AX + B$, где A - матрица, X и B - векторы [4]. Вычисления проводятся по формулам

$$x_i^{(k)} = b_i + \sum_{j=1}^{n} a_{ij} x_j^{(k-1)}, \quad i = 1,2,\ldots,n, \qquad (I)$$

до тех пор, пока не будет достигнута требуемая точность

$$| x_i^{(k)} - x_i^{(k-1)} | < \varepsilon, \quad i = 1,2,\ldots,n. \qquad (2)$$

Ниже представлены программы для решения данной задачи соответственно на одной и L машинах. Первая программа (SMCT8) записана на ФОРТРАНе, вторая (SSCT8) - на ОВС-ФОРТРАНе. Последняя программа выполняется в каждой из L машин и является ветвью параллельной программы из L идентичных ветвей. За каждой ветвью закреплен относительный номер i (i = 1,2,...,L), который заранее сообщается диспетчеру элементарной машины вместе с информацией о числе машин L, принимающих участие в решении данной задачи. Номер ветви определяется номером машины, на которой она реализуется.

Последовательная программа
для решения системы линейных уравнений

```
0001          PROGRAM SMCT8
0002          DIMENSION A(8,8),B(8),X(8),G(8)
0003      334 FORMAT (16HЧИСЛО ИТЕРАЦИЙ I4)
0004      305 FORMAT (8(F8.5,2X))
0005          READ (5,*)N
0006          READ (5,*)((A(I,J),J=1,N),I=1,N)
0007          READ (5,*)(B(I),I=1,N)
0008          READ (5,*)E,ITZ
0009       67 DO 70 I=1,N
0010       70 X(I)=B(I)
0011          ITS=0
0012       75 ITS=ITS+1
0013          DO 80 I=1,N
0014       80 G(I)=X(I)
0015          DO 100 I=1,N
0016          P=0.
0017          DO 90 J=1,N
0018       90 P=P+A(I,J)*G(J)
0019      100 X(I)=B(I)+P
0020          DO 110 I=1,N
0021          P=X(I)-G(I)
0022          P=ABS(P)
0023          IF (P-E) 110,120
0024      110 CONTINUE
0025          GO TO 150
0026      120 IF(ITZ-ITS) 150,75,75
0027      150 WRITE (6,334)ITS
0028          WRITE (6,305)(X(I),I=1,N)
0029      450 STOP
0030          END
0031          END.
```

Ветвь параллельной программы

для решения системы линейных уравнений

```
0001          PROGRAM SECT8
0002          DIMENSION A(4,8),X(8),B(4),G(4),Y(4)
0003          DIMENSION IS(1),IZAN(1),IALP(1),IBEP(1),IZ(1)
0004      334 FORMAT(16H ЧИСЛО ИТЕРАЦИЙ I4)
0005      305 FORMAT(8(F8.5,2X))
0006          READ(5,*)N
0007        7 N7=2*N
0008          CALL RWDKL (N,IS,IZAN,IALP,IBEP,IZ)
0009          IGAN=IZAN
0010          READ(5,*)((A(I,J),J=1,N),I=1,IGAN)
0011          READ(5,*)(B(I),I=1,IGAN)
0012       66 READ(5,*)E,IT2
0013       74 CALL RWDIO
0014          CALL CHEKC(4,N7,B,X)
0015          IT8=0
0016       75 IT8=IT8+1
0017          DO80 J=1,IGAN
0018          I=J+I2-1
0019       80 G(J)=X(I)
0020          DO 100 I=1,IGAN
0021          P=0.
0022          DO 90 J=1,N
0023       90 P=P+A(I,J)*X(J)
0024      100 Y(I)=B(I)+P
0025          CALL CHEKC(6,N7,Y,X)
0026          P=1.
0027          DO 110 I=1,IGAN
0028          P=Y(I)-G(I)
0029          P=ABS(P)
0030          IF(P-E)110,120
0031      110 CONTINUE
0032          P=-1
0033      120 CALL OCP(5,P)
0034          GO TO 150
0035          IF(IT2-IT8)150,75,75
0036      150 WRITE(6,334)IT8
0037          WRITE(6,305)(X(I),I=1,N)
0038      450 STOP
0039          END
0040          END.
```

В процессе выполнения программ SECT 8 и SECT 8 выделяются следующие основные этапы: ввод исходных данных, присвоение начальных значений, вычисления по формуле (I), проверка достижения заданной точности и вывод результатов.

Проследим выполнение указанных этапов в каждой из программ.

В в о д и с х о д н ы х д а н н ы х. Исходными данными являются: порядок системы уравнений ($N \le 8$)[*], матрица A , вектор

[*] Ограничение значения N зависит от свойств языка ФОРТРАН, в котором отсутствует динамическое распределение памяти. Естественно, если в операторе 2 указать бо́льшую размерность массивов, то и значение N может быть бо́льшим.

B , требуемая точность E и верхняя граница числа итераций IT2 . Они вводятся операторами 5-8 в программе ЯИСТ8 и операторами 6, I0-I3 в программе ЯВСТ8 . Во втором случае матрица A и вектор B подготовлены таким образом, что ветвь с номером m , меньшим или равным остатку от деления N на L, вводит $g_m=\left[\frac{N}{L}\right]+1$ строк матрицы A и g_m элементов вектора B , а для ветви с номером, бо́льшим этого остатка, $g_m=\left[\frac{N}{L}\right]$.Первая ветвь вводит первые g_m строк матрицы A и первые g_m элементов вектора B ,вторая ветвь вводит очередные g_m строк и элементов и т.д. Номер строки (I2) ,являющейся в данной ветви начальной,определяется подпрограммой CALL MYNUL (оператор 8), которая использует информацию о порядке системы уравнений (N), а также информацию диспетчера о номере данной ветви и числе L. Кроме того,эта подпрограмма определяет параметры I2AM = g_m , I2LP = $\left[\frac{N}{L}\right]$ и INMT = N - $\left[\frac{N}{L}\right]\cdot L$.

П р и с в о е н и е н а ч а л ь н ы х з н а ч е н и й. Начальными значениями $x_i^{(0)}$ $(i=1,\ldots,N)$ являются элементы вектора B . В программе ЯИСТ8 они присваиваются элементам массива X посредством N повторений цикла (операторы 9,I0). В программе ЯВСТ8-посредством трансляционно-циклического обмена (оператор I4), при котором все L ветвей поочередно, начиная с первой,передадут введенные g_m элементов вектора B в остальные ветви для соответствующих частей массива X .

В ы ч и с л е н и я п о ф о р м у л е (I). Вначале запоминаются значения x_i на предыдущей итерации в массиве G .В программе ЯИСТ8 запоминаются все N значений (операторы I3,I4), а в программе ЯВСТ8 каждая ветвь запоминает g_m значений x_i (операторы I7-I9), где i = I2,...,(I2 + g_m -1). Далее выполняются вычисления по формуле (I). В программе ЯИСТ8 определяются N значений x_i (операторы I5-I9) и запоминаются в массиве X $(i=1,\ldots,N)$. В программе ЯВСТ8 определяются g_m значений x_i (операторы 20-24) и запоминаются в массиве Y. Затем с помощью трансляционно-циклического обмена (оператор 25) эти значения собираются в каждой ветви в массиве X .

П р о в е р к а д о с т и ж е н и й з а д а н н о й т о ч н о с т и. Решение получено при выполнении неравенства (2) для всех x_i $(i=1,2,\ldots,N)$. При невыполнении этого неравенства хотя бы для одного x_i производится очередная итерация, если чис-

ло проведенных итераций (IT8) не превышает заданное (IT2) . В программе SMST8 производится N сравнений (операторы 20–26). В программе SSST8 – g_a сравнений (операторы 26–32), затем ветви выполняют обобщенный условный переход (оператор 33). В случае отрицательного значения обобщенного условия $P = \bigcap_i P_i$ $(i=1,\dots,L)$ управление получает этап выдачи результата, иначе производится очередная итерация, если IT8 ≤ IT2 .

В ы в о д р е з у л ь т а т о в. Операторы 27, 28 в программе SMST8 и операторы 36, 37 в программе SSST8 производят выдачу на печать сообщения о числе проведенных итераций и значений x_i $(i=1,\dots,N)$. В программе SSST8 можно задать выдачу массива Y, тогда каждая ветвь напечатает g_a значений x_i $(i = I2,\dots,I2+g_a-1)$.

Проанализировав каждый из этапов, можно сделать вывод, что время выполнения программы SSST8 уменьшается примерно в L раз по сравнению с временем выполнения программы SMST8 в основном за счет сокращения числа повторений циклов при счете по формулам (I) и (2).

Отметим, что программа SSST8 настраивается на число ветвей $L \le N$ как на параметр, в частности, при L=1 она будет выполняться на одной машине. Такая возможность обеспечивается особенностью реализации операторов групповых взаимодействий [2,3], в частности трансляционно-циклического обмена (операторы I4,25) и обобщенного условного перехода (оператор 33) в программе SSST8 . Кроме того, подпрограмма MYDKL (оператор 8) вычисляет необходимые параметры в соответствии со значением L.

2. Обращение матрицы методом пополнения

Матрица A, обратная исходной матрице W', выполняется за N шагов [4] по рекуррентной формуле

$$\left.\begin{aligned} \alpha_{ij}^{(k)} &= \alpha_{ij}^{(k-1)} - \frac{\alpha_{ik}^{(k-1)}}{1+\alpha_k^{(k)}} \cdot \alpha_j^{(k)} ; \\ \alpha_j^{(k)} &= z_{kj} + \sum_{i=1}^{k-1} w_{ki} \cdot \alpha_{ij}^{(k-1)} ; \\ z_{kj} &= \begin{cases} 0 & \text{при } j < k, \\ w_{kj} & \text{при } j \ge k, \end{cases} \end{aligned}\right\} \qquad (3)$$

здесь $k = 1,2,\ldots,N$; $i = 1,2,\ldots,k$; $j = 1,2,\ldots,N$; N – порядок матрицы W'; w_{ij} – элементы матрицы $W = W' - E$ (E – единичная матрица); $\alpha_{ij}^{(0)}$ – элементы единичной матрицы.

Для решения данной задачи на подсистеме из L элементарных машин рассмотрим программу OCT50.

Ветвь параллельной программы
для решения задачи обращения матрицы

```
0001          PROGRAM OCT50
0002          DIMENSION W(25,50),A(50,25),E(50),W1(50),
0003         .   IS(1),IZAM(1),IALP(1),IBNT(1),IZ(1)
0004      305 FORMAT(8(F8.5,2X))
0005          READ(5,*)N
0006          CALL MODEL(N,IS,IZAM,IALP,IBNT,IZ)
0007        7 N7=2*N
0008          IGAM=IZAM
0009          READ(5,*)((W(I,J),J=1,N),I=1,IGAM)
0010       55 CALL RNDIO
0011          K=0
0012          N1=0
0013       60 N1=N1+1
0014          JNK=IALP
0015          IF(IBNT-N1)80,70,70
0016       70 JNK=JNK+1
0017       80 K=K+1
0018          IK=K-1
0019          K5=2*K
0020          IP(N1-IS)100,90,100
0021       90 IP1=K-IZ+1
0022          DO 91 I=1,IK
0023       91 E(I)=A(I,IP1)
0024          E(K)=1
0025          DO 92 I=1,N
0026       92 W1(I)=W(IP1,I)
0027      100 CALL RMKC(1,N1,K5,E,E)
0028      105 CALL RMKC(2,N1,N7,W1,W1)
0029          P=0.
0030          DO 130 I=1,K
0031      130 P=P+W1(I)*E(I)
0032          P=P+1.0
0033          IP(P)150,140,150
0034      140 STOP 177
0035      150 DO 160 I=1,K
0036      160 E(I)=-E(I)/P
0037          IZ1=IZ
0038          DO 200 J=1,IGAM
0039          P=0.
0040          A(K,J)=0
0041          IP(IZ1-K)164,162,162
0042      162 P=W1(IZ1)
0043      164 DO 170 I=1,IK
0044      170 P=P+W1(I)*A(I,J)
0045          IP(IZ1-K)172,171,172
0046      171 A(K,J)=1.0
```

```
0047      172 IZ1=IZ1+1
0048          DO 180 I=1,K
0049      180 A(I,J)=A(I,J)+B(I)*P
0050      200 CONTINUE
0051          IF(K-E)210,230
0052      210 JEK=JEK-1
0053          IF(JEK)60,60,80
0054      230 WRITE(6,305)((A(I,J),J=1,IQAM),I=1,N)
0055          STOP
0056          END
0057          END.
```

Аналогично программе SSCT8, она является ветвью параллельной программы из L идентичных ветвей и при L = 1 может выполняться на одной машине. Исходными данными для OCTSO являются элементы матрицы W, которая распределена по строкам на L ветвей по g_m (m = = 1,...,L) строк в каждой. Число g_m и другие необходимые параметры определяются, как и в SSCT8 , с помощью подпрограммы RVINKL (оператор 6). Результирующая матрица A распределена по столбцам, и элементы этих столбцов вычисляются по формуле (3) на любом шаге k (k=1,...,N) в каждой ветви. Благодаря такому распределению, на любом шаге k в каждой ветви имеются необходимые значения элементов указанных столбцов,полученные на (k-1) шаге,а значения элементов k-го столбца матрицы A и k-й строки матрицы W располагаются в одной из ветвей (называемой передающей). Эти значения принимаются в остальные ветви из передающей с помощью трансляционного обмена (операторы 27,28). Очевидно, что первые g_1 шагов передающей представляет первую ветвь, очередные g_2 шагов передающей - вторую ветвь, наконец, последние g_L шагов - L-ю ветвь.

Поясним выполнение программы в каждой ветви после ввода исходных данных. Параметр M1 (оператор I3) принимает значение номера передающей ветви, а параметр JEK (операторы I4-I6) - значение количества шагов, в течение которых ветвь M1 будет передающей.

Каждая ветвь, сравнивая (оператор 20) параметр I8 (т.е. свой номер) с параметром M1, определяет, является ли она передающей. При I8 ≠ M1 ветви переходят на оператор 27. При I8 = M1 производится перенос (операторы 2I-26) в массивы B и W1 соответственно элементов k-го столбца матрицы A и k-й строки матрицы W,затем управление получает оператор 27. С помощью трансляционного обмена (операторы 27,28) происходит передача элементов массивов B и W1 во все ветви.

Далее в каждой ветви вычисляется (операторы 29-33) знаменатель $p = 1 + a_k^{(k)}$. При p=0 происходит останов (оператор 34) во всех ветвях. При $p \neq 0$ вычисляются (операторы 35,36) множители

$\frac{\alpha_{ik}^{(k-1)}}{p}$. Наконец (операторы 37-50), вычисляются k элементов каждого из q_m столбцов (m=1,2,...,L).

Затем при k<N (операторы 51-53) управление передается на выполнение очередного шага. При этом номер передающей ветви не меняется, если значение параметра JEK отлично от нуля, иначе - увеличивается на единицу. При k=N происходит выдача результата.

3. Параллельная программа внутренней сортировки

Имеется массив из N неотрицательных чисел, которые необходимо расположить в порядке убывания или возрастания [5]. Предлагается параллельная программа внутренней сортировки, алгоритм которой создан на основе табличного метода. Этот метод является одним из самых быстрых (порядка N сравнений), но требует значительного объема дополнительной памяти.

Суть его в следующем. В оперативной памяти строится таблица, размер которой не меньше величины максимального из сортируемых чисел. Затем за N сравнений таблица заполняется так, что значение i-го элемента таблицы равно количеству чисел величины i-1 в числовом массиве. После заполнения таблицы происходит ее "распаковка", т.е. таблица просматривается сверху донизу, начиная с первого элемента, и на месте исходного массива последовательно записывается столько нулей, сколько содержится первых элементов таблицы, затем столько единиц, сколько содержится вторых элементов и т.д. до последней строки таблицы.

Основываясь на этом методе, параллельный алгоритм заключается в следующем: исходный массив чисел делится между L ветвями параллельной программы с идентичными ветвями; в каждой ветви освобождающаяся память "чистится" и выделяется под таблицу; затем происходит заполнение таблиц, рассылка их трансляционным обменом во все ветви и слияние таблиц, т.е. сложение строк с одинаковыми относительными номерами. После этого каждая ветвь "распаковывает" свою часть таблицы.

Ветвь параллельной программы
для внутренней сортировки массива чисел

```
0001  FIN,B
0002        PROGRAM PSORT
0003      1 COMMON IW(20),M(7000),IT(2050),N,NU
0004        CALL NUM(NU)
0005        I8=2050
```

```
0006          DO 50 I=1,I8
0007       50 IT(I)=0
0008          CONTINUE
0009       61 CALL SNR(5,0)
0010          DO 70 I=1,N
0011          IR=M(I)+1
0012       70 IT(IR)=IT(IR)+1
0013          CALL RMXC(2,1,I8,IT,M)
0014          CALL RMXC(3,2,I8,IT,M)
0015       90 DO 110 I=1,I8
0016      110 IT(I)=IT(I)+M(I)
0017      120 CALL SNR(5,0)
0018          K=0
0019          IK=0
0020          DO 130 I=1,I8
0021          IF(IT(I)-1)130,140
0022      140 IR=IT(I)
0023          DO 150 J=1,IR
0024          IF(K-N)160,160,170
0025      170 IF(1-NU)180,190
0026      180 IK=K+1
0027      160 K=K+1
0028          IK1=K-IK
0029      150 M(IK1)=I-1
0030      130 CONTINUE
0031      190 END
0032          END.
```

В приведенной параллельной программе, использующей описанный метод для $L = 2$ и $N \leq 14000$, величина максимального числа не превышает 2^{11}. Операторы 10–12 производят заполнение таблиц в ветвях, операторы 13–16 – их слияние, а операторы 18–30 – "распаковку" чисел. Подпрограмма NUM определяет номер ветви.

4. Организация продолжения вычислений при отказе одной из элементарных машин подсистемы

Одной из задач высоконадежных (живучих) параллельных вычислений является автоматическое обнаружение неисправной элементарной машины (ЭМ) подсистемы и перенастройка программы вычислений на новое число ЭМ, при этом используется программная избыточность записанного алгоритма.

Приведенная ниже параллельная (р–) программа позволяет автоматически диагностировать отказ одной из двух ЭМ и продолжать вычисления на исправной машине до тех пор, пока не будет запущена вторая. Схема (операторы 20–34) высоконадежных вычислений включена в программу (операторы 4–19), ветви которой выполняют печать четных и нечетных чисел с соответствующими программными таймерами, имитирующими процесс вычислений. Ветви р–программы идентичны.

Ветвь параллельной программы, не прекращающей свои вычисления при отказе одной из ЭМ подсистемы

```
0001  FTN,R,L
0002        PROGRAM FTEST
0003        DIMENSION JA(2),ION(2), IWRT
0004        CALL NUM(IWRT)
0005        IF(1-IWRT)10,20
0006     10 JA(1)=0
0007        JA(2)=1
0008        GOTO 30
0009     20 WRITE(2,430)
0010        READ(1,*)K
0011        JA(1)=1
0012        JA(2)=0
0013     30 CALL EXEC(1,1,K,K)
0014     35 DO 50 I=1,K
0015        WRITE(6,400)JA(1)
0016        CALL ENDIO
0017        DO 40 J=1,32000
0018     40 CONTINUE
0019     50 JA(1)=JA(1)+2
0020     60 CALL SSW
0021        CALL RPT1(1)
0022        GOTO 60
0023        CALL RPT2(1)
0024        GOTO 60
0025        CALL SSW(1,1)
0026        GOTO 70
0027        WRITE(6,420)
0028        CALL EXEC(2,1,2,JA,ION)
0029        CALL EXEC(3,2,2,JA,ION)
0030        IF(JA(1)-ION(2))90,100
0031     90 JA(1)=ION(2)
0032    100 IF(JA(2)-ION(1))110,120
0033    110 JA(2)=ION(1)
0034    120 GOTO 35
0035     70 DO 75 I=1,30000
0036        DO 75 J=1,K
0037     75 CONTINUE
0038    200 DO 180 I=1,K
0039        WRITE(6,410)JA
0040        CALL ENDIO
0041        DO 170 J=1,32000
0042    170 CONTINUE
0043        JA(1)=JA(1)+2
0044    180 JA(2)=JA(2)+2
0045        GOTO 200
0046    400 FORMAT(I8)
0047    410 FORMAT(2I8)
0048    420 FORMAT( "ОБМЕН")
0049    430 FORMAT( "ВВЕДИТЕ ШАГ ПРОВЕРКИ" /I*)
0050        END
0051        END*
```

Выполнение ветви р-программы начинается с присвоения начальных значений массиву JA в зависимости от номера ветви, который определяется функцией NUM , входящей в состав ОВС-языков [3]. Первая ветвь назначается на печать нечетных чисел, вторая - четных. Число итераций K (число отпечатанных чисел), после которого необходимо проверить исправность ЭМ подсистемы, вводится через первую ветвь и передается во вторую операторами 9,10,13. Вторая ветвь ожидает результат ввода значения K при выполнении оператора 13.

После выполнения основной подпрограммы (операторы 14-19), т.е. после печати первой ветвью K нечетных чисел, второй K четных, выполняется оператор 20. Его выполнение приводит элементарную машину в начальное системное состояние ("обнуляет" внутренние семафоры и выдает команду сброса в модуль межмашинной связи). Операторы 21-24 подготавливают системную информацию на случай сбоя или отказа при выполнении межмашинного обмена операторами 28,29. Наличие сбоя в вычислениях или отказа ЭМ приводит к автоматическому перезапуску программ ветвей с метки 60. Оператор 25 задает синхронизацию ветвей с возможностью ответвления на вспомогательные вычисления по метке 70.

Признаком исправности ЭМ является выполнение процессором оператора 25 при условии, что модуль межмашинной связи, подключенный к ней, исправен. Выполнение оператора 25 во всех ветвях р-программы приводит к прерыванию по синхронизации. Все ветви одновременно продолжают выполнение своих программ с оператора 27. Обмены данными, выполняемые операторами 28,29, осуществляются для взаимной информации ветвей о результатах своих вычислений. Операторы 30-33 выбирают информацию о последнем шаге вычислений, т.е. максимальные значения из отпечатанных четных и нечетных чисел. Возврат на нормальные вычисления осуществляется во всех ветвях оператором 34.

Таким образом, после каждых K итераций любая ветвь информирована о результатах вычислений в другой ветви и может продолжить за нее вычисления (если произойдет сбой или отказ) с предыдущего шага.

Отказ одной из ЭМ определяется другой ЭМ по переполнению программного таймера (операторы 35-37), который служит для согласования разбега ветвей по основным вычислениям. Перестройка вычислений осуществляется операторами 38-45.

Приведенная схема параллельной программы выявляет все случаи отказов:

I. При отказе ЭМ до выполнения синхронизации (оператора 25) в исправной ЭМ произойдет переполнение таймера и программа ветви перестроится на меньшее число ЭМ.

2. При отказе ЭМ во время обмена в исправной ЭМ не будет выдана готовность по межмашинной связи и операционная система передаст управление на метку 60. Далее схема диагностики работает, как и в п.I.

3. При отказе модуля межмашинной связи обе ЭМ считают друг друга отказавшими и каждая из них выполняет вычисления за двоих, т.е. вычисления дублируются.

После восстановления отказавшей ЭМ программа в ней запускается с оператора 20. Выполнение ее оператора 25 приводит к прерыванию работающей ЭМ, которая сообщит результаты своих вычислений восстановленной ЭМ. Каждая ветвь р-программы возвращается самостоятельно к выполнению основной программы, осуществляющей счет при исправных ЭМ.

Приведенная схема организации живучих вычислений может быть включена в любую параллельную программу, содержащую указанную избыточность в программном описании алгоритма вычислений.

Литература

I. МИРЕНКОВ Н.Н. МИНИМАКС - вычислительная система коллективного пользования.-Вычислительные системы. (Вопросы теории и построения вычислительных систем.) Вып.60, 1974, с. 115-129.

2. КЕРБЕЛЬ В.Г., КОЛОСОВА Ю.И., КОРНЕЕВ В.Д., МИРЕНКОВ Н.Н., ЩЕРБАКОВ Е.В. Средства программирования системы МИНИМАКС.- Вычислительные системы. (Вопросы теории и построения вычислительных систем.) Вып.60, 1974, с. 143-152.

3. КЕРБЕЛЬ В.Г., КОЛОСОВА Ю.И., КОРНЕЕВ В.Д., КРЫЛОВ З.Г., МИРЕНКОВ Н.Н. Программное обеспечение системы МИНИМАКС.-Новосибирск. Б.и.,1979.-48 с.-(Препринт / ВМ СО АН СССР; ОВС-09).

4. ФАДДЕЕВ Д.К., ФАДДЕЕВА В.Н. Вычислительные методы линейной алгебры. -М.: 1960. -656 с.

5. КНУТ Д. Искусство программирования для ЭВМ. т.3. Сортировка и поиск. -М.: Мир, 1978. -843 с.

Поступила в ред.-изд.отд.
15 мая 1979 года