Automatic Parallelization and Manual Improvements of the Perfect Club Program MDG on Alliant FX/80, FX8, and Cedar *Working Report*

Rudolf Eigenmann

June 1992

Contents

*

1 Overall timings

	Cedar	FX/80	speed improvements on FX8 / Cedar over			
			serial	vector	best-FX8	best-FX80
serial	4134 (4047)	2432 (2339)	1			
vector	3200 (3118)	1905 (1833)	1.3	1		
kap/fx8	4314 (4208)	2906 (2799)	.95	.74		
fx8[0]-opt	618 (598)	447 (428)	6.7	5.2	1	1
fx8-opt+global	832 (808)		5.0	3.8		
Cedar-opt	207 (198)		20.0	15.5	3.0	2.2

2 Optimization for Alliant FX/80

Loop by loop profiles: the table shows the total amount of cpu time spent in the 10 most significant loops

loop	serial	auto-parallelized	manually improved
INTERF_do1000	452.628967	518.781372	75.657516
POTENG_do2000	31.678848	49.766144	6.153000
PREDIC_do1000	5.336384	6.361679	6.458436
CORREC_do1000	1.237518	1.553776	1.573189
INTRAF_do1000	0.856993	0.849568	0.867073
MDG_do101	0.181290	0.029660	0.030420
INTRAF_do2000	0.160664	0.042052	0.042699
KINETI_do100	0.097355	0.025455	0.026071
INTERF_do2000	0.096136	0.107460	0.051739

2.1 Source of the most important loop

The loop 1000 of subroutine INTERF spends 90% of the sequential program execution time. 8% of the execution time is spent in loop 2000 of subroutine poteng. The following excerpts shows some of the loop characteristics. The line numbers indicate the sizes of the cuts.

```
  3        SUBROUTINE INTERF(X,Y,Z,FX,FY,FZ,XM,YM,ZM,VIR)
...
 18        DIMENSION XL(14),YL(14),ZL(14),RL(14),RS(14),FF(14),GG(14)
...
 20        IW1=1
 21        IWO=2
 22        IW2=3
 23        DO 1000 I=1,NMOL1
 24           JW1=IW1
 25           JWO=IWO
 26           JW2=IW2
 27           DO 1100 J=I+1,NMOL
 28              JW1=JW1+NATOMS
 29              JWO=JWO+NATOMS
 30              JW2=JW2+NATOMS
 31              CALL CSHIFT(X(IW1),X(JW1),XM(I),XM(J),XL,BOXH,BOXL)
...
 35              DO 1110 K=1,9
 36                 RS(K)=XL(K)*XL(K)+YL(K)*YL(K)+ZL(K)*ZL(K)
 37  1110        IF(RS(K).GT.CUT2) KC=KC+1
...
 48              VIR=VIR+FF(K)*RS(K)
...
 58              VIR=VIR+FF(10)*RS(10)
```

```
...
 71     20       DO 1150 K=1,14
 72   1150          GG(K)=FF(K)*XL(K)
...
 76              FX(IWO)=FX(IWO)+G110+GG(11)+GG(12)+C1*G23
 77              FX(JWO)=FX(JWO)-G110-GG(13)-GG(14)-C1*G45
...
119   1100    CONTINUE
120           IW1=IW1+NATOMS
121           IWO=IWO+NATOMS
122           IW2=IW2+NATOMS
123   1000 CONTINUE


 2         SUBROUTINE POTENG(X,Y,Z,XM,YM,ZM,POTA,POTR,PTRF)
...
 8         IMPLICIT DOUBLE PRECISION(A-H,O-Z)
 9         COMMON /MDVAR/ TEMP,RHO,TSTEP,BOXL,BOXH,CUTOFF,CUT2,NMOL,NORDER,
 10        *                NATMO,NATMO3,NMOL1
 11         COMMON /WATER/ OMAS,HMAS,WTMOL,ROH,ANGLE,FHM,FOM,ROHI,ROHI2,NATOMS
 12         COMMON /WWPOT/ QQ,A1,B1,A2,B2,A3,B3,A4,B4,AB1,AB2,AB3,AB4,C1,C2,
 13        *                QQ2,QQ4,REF1,REF2,REF4
 14         COMMON /FRCNST/ FC11,FC12,FC13,FC33, FC111,FC333,FC112,FC113,
 15        *                 FC123,FC133, FC1111,FC3333,FC1112,
 16        *                 FC1122,FC1113,FC1123,FC1133,FC1233,FC1333
 17         DIMENSION X(1),Y(1),Z(1),XL(14),YL(14),ZL(14),RL(14),RS(14)
 18         DIMENSION XM(1),YM(1),ZM(1)
...
 69         DO 2000 I=1,NMOL1
 70         JW1=IW1+NATOMS
 71         JWO=JW1+1
 72         JW2=JW1+2
 73         JMIN=I+1
 74         DO 2100 J=JMIN,NMOL
 75         CALL CSHIFT(X(IW1),X(JW1),XM(I),XM(J),XL,BOXH,BOXL)
 76         CALL CSHIFT(Y(IW1),Y(JW1),YM(I),YM(J),YL,BOXH,BOXL)
 77         CALL CSHIFT(Z(IW1),Z(JW1),ZM(I),ZM(J),ZL,BOXH,BOXL)
 78         KC=0
 79         DO 2110 K=1,9
 80         RS(K)=XL(K)*XL(K)+YL(K)*YL(K)+ZL(K)*ZL(K)
 81    2110 IF(RS(K).GT.CUT2) KC=KC+1
 82         IF(KC.EQ.9) GO TO 100
 83        DO 2120 K=1,9
 84        IF(RS(K).GT.CUT2) GOTO 10
 85        RL(K)=SQRT(RS(K))
 86        GO TO 2120
 87     10 RL(K)=CUTOFF
 88        RS(K)=CUT2
 89   2120 CONTINUE
 90        POTR=POTR-QQ2/RL(2)-QQ2/RL(3)-QQ2/RL(4)-QQ2/RL(5)
 91       *         +QQ /RL(6)+QQ /RL(7)+QQ /RL(8)+QQ /RL(9)
 92       *         +QQ4/RL(1)
 93        PTRF=PTRF-REF2*RS(1)-REF1*((RS(6)+RS(7)+RS(8)+RS(9))*0.5D0
 94       *                           -RS(2)-RS(3)-RS(4)-RS(5))
 95        IF(KC.GT.0) GOTO 100
 96        DO 2130 K=10,14
 97   2130 RL(K)=SQRT(XL(K)*XL(K)+YL(K)*YL(K)+ZL(K)*ZL(K))
 98        POTR=POTR+A1* EXP(-B1*RL(10))
 99       *         +A2*(EXP(-B2*RL( 6))+EXP(-B2*RL( 7))
 100       *                           +EXP(-B2*RL( 8))+EXP(-B2*RL( 9)))
 101       *         +A3*(EXP(-B3*RL(11))+EXP(-B3*RL(12))
 102       *                           +EXP(-B3*RL(13))+EXP(-B3*RL(14)))
 103       *         -A4*(EXP(-B4*RL(11))+EXP(-B4*RL(12))
 104       *                           +EXP(-B4*RL(13))+EXP(-B4*RL(14)))
```

```
105     100 JW1=JW1+NATOMS
106         JWO=JWO+NATOMS
107         JW2=JW2+NATOMS
108    2100 CONTINUE
109         IW1=IW1+NATOMS
110         IWO=IWO+NATOMS
111         IW2=IW2+NATOMS
112    2000 CONTINUE
```

2.2 Automatic parallelization

When automatically optimizing this code, the Restructurer parallelizes the innermost loops, such as do-1110 and do-1150 in subroutine INTERF. The outer loops INTERF_ do-1000 and POTENG_do200 are not recognized as parallel loops. Subroutine expansion has been applied. It successfully expands subroutine CSHIFT (containing about 20 lines). This does not affect the optimization, though.

The vector-concurrent version runs slower than serial. Concurrent-only runs 25% faster. The same speedup results from vectorized-only. This is due to the low iteration count in all inner loops of less than 10. Kap stripmines the loops which introduces but overhead. Kap does not look at iteration counts, even when they are known.

2.3 Manual optimization

2.3.1 Subroutine INTERF

Loop do-1000 can be executed in concurrent mode. Among the steps needed for this are:

- There are three induction variables IW1, IWO, IW2 whose increment 'NATOMS' on lines 120..122 is not known in this subroutine. Its constant value '3' can be found by looking at the whole program. There are no data dependences across loop iterations caused by these index variables.
- All arrays declared on source line 18 are read and written by all loop iterations. They are always written before they are used, thus they can be declared local to the concurrent do-1000 loop (line 81 of the transformed code). This is analogue to a technique sometimes called 'array expansion' where an array dimension is added, so each loop iteration has its own variant of the array.
- Many data dependences that inhibit automatic parallelization are caused by the accumulation patterns found on source lines 48/58/76/77. The variables are only referenced for summing values computed in the do-1100 loop iterations. To break these dependences we declare local variables (line 50) that hold partial sums (lines 149, 154, 187) which are added to the original variable in a critical section at the end of the loop. Some of these values need be added in every do-1100 iteration (line 234). Others can be summed in the outer loop (line 251..253).

```
 18        SUBROUTINE INTERF(X,Y,Z,FX,FY,FZ,XM,YM,ZM,VIR)
...

 49        CDOALL 1000 I=1,NMOL1
...
 81        DOUBLE PRECISION XL(14), YL(14), ZL(14), RL(14), RS(14), FF(14), GG(14)
 82        double precision virl, fxi1, fxio, fxi2, fxj1, fxjo, fxj2
...
 87        IW1=(I-1)*NATOMS +1
...
 90        DO 1100 J=I+1,NMOL
 91        JW1 = (J-1) * NATOMS + 1
...
104        XL(11) = Xloc(IW1+1) - Xloc(JW1)
...
```

```
143        RS(1:$9) = XL(1:$9) * XL(1:$9) + YL(1:$9) * YL(1:$9) + ZL(1:$9) * ZL(1:$9)
144   1110 KC = KC + COUNT (MASK=RS(1:$9) .GT. CUT2)
...
149        VIRL=VIRL+FF(1)*RS(1)
...
154        VIRL=VIRL+FF(K)*RS(K)
...
183   1150 GG(1:$14) = FF(1:$14) * XL(1:$14)
...
187        FXIO=+G110+GG(11)+GG(12)+C1*G23
188        FXJO=-G110-GG(13)-GG(14)-C1*G45
...
229        call set_qlock(lock)
...
234        fx(jwo) = fx(jwo) + fxjo
...
239        call clear_qlock(lock)
240   1100 CONTINUE
241        call set_qlock(lock)
242        fx(iwo) = fx(iwo) + fxio
...
252        vir = vir + virl
253        call clear_qlock(lock)
254   1000 CONTINUE
```

2.3.2 Subroutine POTENG

The patterns in this loop are very similar to loop INTERF_do1000. The first two transformations are the same: Induction variables are replaced and the arrays declared on line 17 are declared local to loop 2000.

There is again an accumulation pattern for the variables POTR and PTRF. Here, local variables POTRL and PTRFL are declared. They accumulate local sums. The values are updated on the original values at the end of the loop 2000 under a AWAIT/ADVANCE pair. This type synchronization is possible since here we have no variables that need be updated in the inner loop 2100. The updates correspond to the ones on lines 250..253 in the INTERF_do100 loop, where ADVANCE/AWAIT synchronization would have beed an alternative. Note, for the inner update (on lines 234 above) we could not have used AWAIT/ADVANCE.

```
3          SUBROUTINE POTENG(X,Y,Z,XM,YM,ZM,POTA,POTR,PTRF)
...
32          CDOACROSS 1000 I=1,NMOL
...
83          POTR=0.0D0
84          PTRF=0.0D0
...
86          CDOACROSS 2000 I=1,NMOL1
87          integer iw1,iwo,iw2,jw1,jwo,jw2,kc
88          INTEGER II1, II2, II3, j, k
89          DOUBLE PRECISION XL(14), YL(14), ZL(14), RL(14), RS(14), potrl, ptrfl
90
91          potrl = 0.0D0
92          ptrfl = 0.0D0
93          IW1 = 1+ (I-1)*NATOMS
94          IWO = IW1 + 1
95          IW2 = IW1 + 2
96          JW1=IW1+NATOMS
97          JWO=JW1+1
98          JW2=JW1+2
99          JMIN=I+1
100          DO 2100 J=I+1,NMOL
101          XL(1) = XM(I) - XM(J)
```

```
      XL(2) = XM(I) - X(JW1)
      XL(3) = XM(I) - X(JW1+2)
      XL(4) = X(IW1) - XM(J)
      XL(5) = X(IW1+2) - XM(J)
      XL(6) = X(IW1) - X(JW1)
      XL(7) = X(IW1) - X(JW1+2)
      XL(8) = X(IW1+2) - X(JW1)
      XL(9) = X(IW1+2) - X(JW1+2)
      XL(10) = X(IW1+1) - X(JW1+1)
      XL(11) = X(IW1+1) - X(JW1)
      XL(12) = X(IW1+1) - X(JW1+2)
      XL(13) = X(IW1) - X(JW1+1)
      XL(14) = X(IW1+2) - X(JW1+1)
      WHERE (ABS (XL(1:$14)) .GT. BOXH) XL(1:$14) = XL(1:$14) - SIGN (
     X  BOXL, XL(1:$14))
    2 CONTINUE
      YL(1) = YM(I) - YM(J)
      YL(2) = YM(I) - Y(JW1)
      YL(3) = YM(I) - Y(JW1+2)
      YL(4) = Y(IW1) - YM(J)
      YL(5) = Y(IW1+2) - YM(J)
      YL(6) = Y(IW1) - Y(JW1)
      YL(7) = Y(IW1) - Y(JW1+2)
      YL(8) = Y(IW1+2) - Y(JW1)
      YL(9) = Y(IW1+2) - Y(JW1+2)
      YL(10) = Y(IW1+1) - Y(JW1+1)
      YL(11) = Y(IW1+1) - Y(JW1)
      YL(12) = Y(IW1+1) - Y(JW1+2)
      YL(13) = Y(IW1) - Y(JW1+1)
      YL(14) = Y(IW1+2) - Y(JW1+1)
      WHERE (ABS (YL(1:$14)) .GT. BOXH) YL(1:$14) = YL(1:$14) - SIGN (
     X  BOXL, YL(1:$14))
    3 CONTINUE
      ZL(1) = ZM(I) - ZM(J)
      ZL(2) = ZM(I) - Z(JW1)
      ZL(3) = ZM(I) - Z(JW1+2)
      ZL(4) = Z(IW1) - ZM(J)
      ZL(5) = Z(IW1+2) - ZM(J)
      ZL(6) = Z(IW1) - Z(JW1)
      ZL(7) = Z(IW1) - Z(JW1+2)
      ZL(8) = Z(IW1+2) - Z(JW1)
      ZL(9) = Z(IW1+2) - Z(JW1+2)
      ZL(10) = Z(IW1+1) - Z(JW1+1)
      ZL(11) = Z(IW1+1) - Z(JW1)
      ZL(12) = Z(IW1+1) - Z(JW1+2)
      ZL(13) = Z(IW1) - Z(JW1+1)
      ZL(14) = Z(IW1+2) - Z(JW1+1)
      WHERE (ABS (ZL(1:$14)) .GT. BOXH) ZL(1:$14) = ZL(1:$14) - SIGN (
     X  BOXL, ZL(1:$14))
    4 CONTINUE
      KC = 0
      RS(1:$9) = XL(1:$9) * XL(1:$9) + YL(1:$9) * YL(1:$9) + ZL(1:$9) *
     X  ZL(1:$9)
 2110 KC = KC + COUNT (MASK=RS(1:$9) .GT. CUT2)
      IF (KC .NE. 9) THEN
      WHERE (RS(1:$9) .LE. CUT2)
      RL(1:$9) = SQRT (RS(1:$9))
   10 OTHERWISE
      RL(1:$9) = CUTOFF
      RS(1:$9) = CUT2
      END WHERE
      POTR1=POTR1-QQ2/RL(2)-QQ2/RL(3)-QQ2/RL(4)-QQ2/RL(5)
     *         +QQ /RL(6)+QQ /RL(7)+QQ /RL(8)+QQ /RL(9)
     *         +QQ4/RL(1)
```

```
      PTRF1=PTRF1-REF2*RS(1)-REF1*((RS(6)+RS(7)+RS(8)+RS(9))*0.5D0
     *                            -RS(2)-RS(3)-RS(4)-RS(5))
      IF (KC .LE. 0) THEN
 2130 RL(10:$5) = SQRT (XL(10:$5) * XL(10:$5) + YL(10:$5) * YL(10:$5) +
     X  ZL(10:$5) * ZL(10:$5))
      POTR1=POTR1+A1* EXP(-B1*RL(10))
     *         +A2*(EXP(-B2*RL( 6))+EXP(-B2*RL( 7))
     *                             +EXP(-B2*RL( 8))+EXP(-B2*RL( 9)))
     *         +A3*(EXP(-B3*RL(11))+EXP(-B3*RL(12))
     *                             +EXP(-B3*RL(13))+EXP(-B3*RL(14)))
     *         -A4*(EXP(-B4*RL(11))+EXP(-B4*RL(12))
     *                             +EXP(-B4*RL(13))+EXP(-B4*RL(14)))
      END IF
      END IF
  100 JW1 = JW1 + NATOMS
      JW0 = JW0 + NATOMS
      JW2 = JW2 + NATOMS
 2100 CONTINUE
      CALL AWAIT (1, 1, POTR,PTRF)
      POTR = ROTR + potrl
      PTRF = PTRF + ptrfl
      CALL ADVANCE (1, POTR,PTRF)

 2000 CONTINUE
        call end_interval( 25 )
      RETURN
      END
```

3 Optimization for Cedar

3.1 Subroutine INTERF

In a first port from FX/8 to Cedar, The cdoall-1000 loop of the Alliant/FX8-optimized code was turned into an xdoall loop. In Cedar, data declared private to a concurrent loop is placed in cluster memory. Since a lot of computation in loop do-1000 operates on privatized arrays, the program takes well advantage of the Cedar memory hierarchy. For example, the computations of the loops 1110 and 1150 are completely localized, i.e. they reference cluster memories only. The *globalization penalty* (see profile in Section 4.5) is 29%, which is relatively low, considering the many scalar accesses. This indicates that data privatization is effective. The initial *spreading overhead* is low, too. This is because the loop body and the number of iterations are large. However, the *spreading speedup* is limited (2.7), which is due to the synchronization in this loop.

Further improvement of this loop was achieved by completely removing the synchronization from the inner loop 1100. This was done by expanding the variables fx, fy, and fz per processor and summing the partial results at the end of the 1100 loop. The additional computation for initializing and summing these variables is less costly than the performance gain due to reduced synchronization. The total time spent in loop interf-1000 got reduced from initially 253 seconds to 170 seconds. The spreading speedup is now 3.8, which is very good. There is also a gain in spreading overhead. This comes from the localization of scalar global variables, which is an additional transformation: the read-only variables **X,Y,Z,XM,YM,ZM** are copied to cluster memory in order to reduce global access cost. The restulting effect is not very pronounced, corresponding to the fact that the initial globalization penalty was not very high (It is unclear how much the copy instructions offset a further gain). The resulting cluster speedup is 3.0.

Profile 4.7 shows performance degradation when using the static scheduling library, as shown in profile Section 4.7. The loop total time goes up from 170 seconds to 226 seconds. This is due to the inner loop which is triangular and thus causes a significant load imbalance.

3.2 Subroutine POTENG

Loop 2000 of subroutine POTENG was transformed into a cdoacross for the alliant/fx8-optimized code. Since there is no xdoacross construct in Cedar Fortran the xdoall construct was used and the synchronization was realized by qlock functions. This synchronization is now unordered, which works as well for the accumulation operations.

This loop is in many respects similar to interf/1000. The globalization penalty is 26%. There is no spreading overhead. The spreading speedup is 4.0. The resulting cluster speedup is a respectable 3.3. There is no synchronization necessary in the inner loop, which explains the good spreading speedup.

There is the same effect using the static scheduling runtime libray as for loop interf/1000. The inner triangular loop causes a load inbalance. The loop timing goes up from 13.4 to 24.5 seconds.

3.3 Subroutine PREDIC

Kap transformed the triply-nested loop predic-1000 into a dotproduct and two outer serial loops.

A number of transformations were applied to in the manually improved version which led to a reduction of the time from 69 to 4 seconds:

- The outer two loops were collapsed. This is beneficial because loop 1000 (originally outermost) has only 6 iterations. The loop is then stripmined into sdoall/cdoall.
- The innermost dotproduct loop is run serially. It has 3 iterations in average.
- variable `V` is copied into a local variable at the beginning of the loop. This solves two problems: First, it breaks the anti dependence on array `V` (The array is overwritten by the loop nest). Second it reduces the globalization penalty of the serial accesses to `V`.

The apparent cluster speedup of 10.6 is only because the manual modifications are not applied to the 1-cluster version.

3.4 Subroutine CORREC

Kap did not recognize the outer loop to be concurrent in the doubly-nested loop CORREC-1000. (Apparently it is the same problem seem in flo52: a temporary scalar inserted by the induction variable pass did not get expanded or privatized, resp.) The outer loop was transformed into a xdoall manually. The loop timing was reduced from 39 to 20 seconds.

The 1-cluster execution of this loop is almost 10 times faster than the 4-cluster run!! This needs to be looked at.

3.5 Subroutine INTRAF

The outer loop of the nest intraf-1000 could be transformed into an xdoall after induction variable substitution and privatization of scalars and a few small arrays (of length 3). The timing is reduced from 16.7 to 1.8 seconds. However there is no spreading speedup. This is another point for further investigation.

4 Profile data

4.1 Nr of iterations

```
  BNDRY_do100    100 AVE:        1029 MIN:   1029 MAX:     1029 TOT:   102900
CNSTNT_do1000      1 AVE:           6 MIN:      6 MAX:        6 TOT:        6
CNSTNT_do1100      6 AVE:           3 MIN:      1 MAX:        6 TOT:       21
CORREC_do1000    100 AVE:        3087 MIN:   3087 MAX:     3087 TOT:   308700
```

```
CORREC_do1100  ***** AVE:           7 MIN:      7 MAX:       7 TOT:  2160900
 CSHIFT_do100  ***** AVE:          14 MIN:     14 MAX:      14 TOT: 273440286
 INITIA_do100      1 AVE:         343 MIN:    343 MAX:     343 TOT:      343
 INITIA_do200      1 AVE:         343 MIN:    343 MAX:     343 TOT:      343
 INITIA_do200    343 AVE:           3 MIN:      3 MAX:       3 TOT:     1029
INITIA_do2000      1 AVE:         343 MIN:    343 MAX:     343 TOT:      343
INITIA_do2000    343 AVE:           3 MIN:      3 MAX:       3 TOT:     1029
 INITIA_do300      1 AVE:        1029 MIN:   1029 MAX:    1029 TOT:     1029
 INITIA_do400      1 AVE:        1029 MIN:   1029 MAX:    1029 TOT:     1029
INTERF_do1000    101 AVE:         342 MIN:    342 MAX:     342 TOT:    34542
INTERF_do1100  34542 AVE:         173 MIN:      3 MAX:     344 TOT:  5993037
INTERF_do1110  ***** AVE:           9 MIN:      9 MAX:       9 TOT: 53315577
INTERF_do1120  ***** AVE:          14 MIN:     14 MAX:      14 TOT: 52131646
INTERF_do1130  ***** AVE:           4 MIN:      4 MAX:       4 TOT: 14894756
INTERF_do1140  ***** AVE:           4 MIN:      4 MAX:       4 TOT:  9801776
INTERF_do1150  ***** AVE:          14 MIN:     14 MAX:      14 TOT: 52131646
INTERF_do1160  ***** AVE:          14 MIN:     14 MAX:      14 TOT: 52131646
INTERF_do1170  ***** AVE:          14 MIN:     14 MAX:      14 TOT: 52131646
INTERF_do2000    101 AVE:         343 MIN:    343 MAX:     343 TOT:    34643
INTRAF_do1000    101 AVE:         343 MIN:    343 MAX:     343 TOT:    34643
INTRAF_do1100  34643 AVE:           3 MIN:      3 MAX:       3 TOT:   103929
INTRAF_do1200  34643 AVE:           3 MIN:      3 MAX:       3 TOT:   103929
INTRAF_do1300  34643 AVE:           3 MIN:      3 MAX:       3 TOT:   103929
INTRAF_do2000    101 AVE:        3087 MIN:   3087 MAX:    3087 TOT:   311787
 KINETI_do100    100 AVE:           3 MIN:      3 MAX:       3 TOT:      300
 KINETI_do110    300 AVE:         343 MIN:    343 MAX:     343 TOT:   102900
    MDG_do100      1 AVE:          24 MIN:     24 MAX:      24 TOT:       24
    MDG_do101      1 AVE:       27783 MIN:  27783 MAX:   27783 TOT:    27783
MDMAIN_do2000      1 AVE:         100 MIN:    100 MAX:     100 TOT:      100
POTENG_do1000     10 AVE:         343 MIN:    343 MAX:     343 TOT:     3430
POTENG_do2000     10 AVE:         342 MIN:    342 MAX:     342 TOT:     3420
POTENG_do2100   3420 AVE:         171 MIN:      1 MAX:     342 TOT:   586530
POTENG_do2110  ***** AVE:           9 MIN:      9 MAX:       9 TOT:  5278770
POTENG_do2120  ***** AVE:           9 MIN:      9 MAX:       9 TOT:  3319002
POTENG_do2130  ***** AVE:           5 MIN:      5 MAX:       5 TOT:  1212810
PREDIC_do1000    100 AVE:           6 MIN:      6 MAX:       6 TOT:      600
PREDIC_do1100    600 AVE:        3087 MIN:   3087 MAX:    3087 TOT:  1852200
PREDIC_do1110  ***** AVE:           3 MIN:      1 MAX:       6 TOT:  6482700
    RAND_do20      1 AVE:           5 MIN:      5 MAX:       5 TOT:        5
    RAND_do50    127 AVE:           4 MIN:      4 MAX:       4 TOT:      508
```

4.2 Profile of total time accumulated by the loops

SUBROUTINE	fx8 versions					cedar versions			
	serial	vector	automatic	manual	manual	first		best	
					+global	1-cl	4-cl	1-cl	4-cl
MDMAIN_do2000	3990.1	3092.7	4167.9	612.94	811.45	857.42	390.46	710.84	216.45
INTERF_do1000	3707.8	2841.5	3791.9	512.62	662.49	683.81	253.18	647.10	169.66
CORREC_do1000	7.4752	4.8532	8.5463	8.4606	18.539	39.614	39.345	2.2551	20.136
POTENG_do2000	272.43	221.53	352.77	43.710	55.070	53.345	13.384	53.504	13.369
PREDIC_do1000	27.774	43.079	43.039	44.017	69.091	69.230	69.089	13.713	4.1596
INTRAF_do1000	8.6096	8.5538	8.1244	8.4214	11.393	16.831	16.720	1.914	1.584
BNDRY_do100	0.4593	0.3521	0.3613	0.3526	0.8803	0.8270	0.8701	0.8615	0.8509
INTRAF_do2000	0.9205	0.1613	0.1621	0.1621	0.2378	0.2375	0.2378	0.2349	0.2528
MDG_do101	0.2014	0.1699	0.2545	0.2506	0.1617	0.2082	0.2098	0.1783	0.1415
KINETI_do100	0.6502	0.2877	0.0946	0.0947	0.1290	0.1289	0.1538	0.1295	0.1295
INITIA_do400	0.0044	0.0013	0.0004	0.0003	0.0005	0.0437	0.1193	0.0530	0.1220
INITIA_do2000	0.0559	0.0564	0.0554	0.0559	0.0903	0.0997	0.1073	0.1007	0.1072
MDG_do100	0.0000	0.0000	0.0000	0.0525	0.0000	0.0512	0.0520	0.0877	0.0885
POTENG_do1000	0.4243	0.1981	0.1546	0.0610	0.0915	0.1203	0.1181	0.0862	0.0863
INITIA_do300	0.0422	0.0469	0.0920	0.0461	0.0531	0.0473	0.0507	0.0573	0.0591
CNSTNT_do1000	0.0029	0.0033	0.0455	0.0027	0.0643	0.0033	0.0029	0.0548	0.0546
INITIA_do200	0.0169	0.0127	0.0016	0.0016	0.0020	0.0236	0.0264	0.0311	0.0239
INITIA_do100	0.0128	0.0053	0.0048	0.0048	0.0075	0.0089	0.0089	0.0089	0.0089
RAND_do50	0.0075	0.0075	0.0076	0.0142	0.0078	0.0078	0.0078	0.0078	0.0078
RAND_do20	0.0000	0.0002	0.0555	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
CSHIFT_do100	------	------	------	------	------	------	------	------	------
INITIA_do1000	------	------	------	------	------	------	------	------	------
INTERF_do2000	0.4331	0.4159	0.4093	0.1526	0.1939	0.1900	0.1809	------	------
MDG_do210	------	------	------	------	------	------	------	------	------
NRMLKT_do300	------	------	------	------	------	------	------	------	------

4.3 Average loop execution times

sorted in alphabetical order

SUBROUTINE	fx8 versions					cedar versions			
	serial	vector	automatic	manual	manual	first		best	
					+global	1-cl	4-cl	1-cl	4-cl
BNDRY_do100	0.0045	0.0035	0.0036	0.0035	0.0088	0.0082	0.0087	0.0086	0.0085
CNSTNT_do1000	0.0029	0.0033	0.0455	0.0027	0.0643	0.0033	0.0029	0.0548	0.0546
CORREC_do1000	0.0747	0.0485	0.0854	0.0846	0.1853	0.3961	0.3934	0.0225	0.2013
CSHIFT_do100	------	------	------	------	------	------	------	------	------
INITIA_do100	0.0128	0.0053	0.0048	0.0048	0.0075	0.0089	0.0089	0.0089	0.0089
INITIA_do1000	------	------	------	------	------	------	------	------	------
INITIA_do200	0.0169	0.0127	0.0016	0.0016	0.0020	0.0236	0.0264	0.0311	0.0239
INITIA_do2000	0.0559	0.0564	0.0554	0.0559	0.0903	0.0997	0.1073	0.1007	0.1072
INITIA_do300	0.0422	0.0469	0.0920	0.0461	0.0531	0.0473	0.0507	0.0573	0.0591
INITIA_do400	0.0044	0.0013	0.0004	0.0003	0.0005	0.0437	0.1193	0.0530	0.1220
INTERF_do1000	36.711	28.134	37.543	5.0754	6.5593	6.7704	2.5067	6.4069	1.6798
INTERF_do2000	0.0042	0.0041	0.0040	0.0015	0.0019	0.0018	0.0017	------	------
INTRAF_do1000	0.0852	0.0846	0.0804	0.0833	0.1128	0.1666	0.1655	0.0189	0.0156
INTRAF_do2000	0.0091	0.0015	0.0016	0.0016	0.0023	0.0023	0.0023	0.0023	0.0025
KINETI_do100	0.0065	0.0028	0.0009	0.0009	0.0012	0.0012	0.0015	0.0012	0.0012
MDG_do100	0.0000	0.0000	0.0000	0.0525	0.0000	0.0512	0.0520	0.0877	0.0885
MDG_do101	0.2014	0.1699	0.2545	0.2506	0.1617	0.2082	0.2098	0.1783	0.1415
MDG_do210	------	------	------	------	------	------	------	------	------
MDMAIN_do2000	3990.1	3092.7	4167.9	612.94	811.45	857.42	390.46	710.84	216.45
NRMLKT_do300	------	------	------	------	------	------	------	------	------
POTENG_do1000	0.0424	0.0198	0.0154	0.0061	0.0091	0.0120	0.0118	0.0086	0.0086
POTENG_do2000	27.243	22.153	35.277	4.3710	5.5070	5.3345	1.3384	5.3504	1.3369
PREDIC_do1000	0.2777	0.4307	0.4303	0.4401	0.6909	0.6923	0.6908	0.1371	0.0415
RAND_do20	0.0000	0.0002	0.0555	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
RAND_do50	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000

4.4 Minumum loop execution times

sorted in alphabetical order

SUBROUTINE		fx8 versions				cedar versions			
	serial	vector	automatic	manual	manual	first		best	
					+global	1-cl	4-cl	1-cl	4-cl
BNDRY_do100	0.0042	0.0035	0.0035	0.0035	0.0082	0.0082	0.0082	0.0082	0.0082
CNSTNT_do1000	0.0029	0.0033	0.0455	0.0027	0.0643	0.0033	0.0029	0.0548	0.0546
CORREC_do1000	0.0731	0.0474	0.0829	0.0821	0.1795	0.3818	0.3818	0.0217	0.0740
CSHIFT_do100	------	------	------	------	------	------	------	------	------
INITIA_do100	0.0128	0.0053	0.0048	0.0048	0.0075	0.0089	0.0089	0.0089	0.0089
INITIA_do1000	------	------	------	------	------	------	------	------	------
INITIA_do200	0.0169	0.0127	0.0016	0.0016	0.0020	0.0236	0.0264	0.0311	0.0239
INITIA_do2000	0.0559	0.0564	0.0554	0.0559	0.0903	0.0997	0.1073	0.1007	0.1072
INITIA_do300	0.0422	0.0469	0.0920	0.0461	0.0531	0.0473	0.0507	0.0573	0.0591
INITIA_do400	0.0044	0.0013	0.0004	0.0003	0.0005	0.0437	0.1193	0.0530	0.1220
INTERF_do1000	36.643	28.091	37.497	5.0462	6.5213	6.7372	2.4555	6.3655	1.6510
INTERF_do2000	0.0039	0.0040	0.0039	0.0013	0.0017	0.0017	0.0017	------	------
INTRAF_do1000	0.0826	0.0824	0.0778	0.0811	0.1098	0.1610	0.1610	0.0178	0.0086
INTRAF_do2000	0.0089	0.0015	0.0015	0.0015	0.0023	0.0023	0.0023	0.0023	0.0023
KINETI_do100	0.0065	0.0027	0.0009	0.0009	0.0012	0.0012	0.0012	0.0012	0.0012
MDG_do100	0.0000	0.0000	0.0000	0.0525	0.0000	0.0512	0.0520	0.0877	0.0885
MDG_do101	0.2014	0.1699	0.2545	0.2506	0.1617	0.2082	0.2098	0.1783	0.1415
MDG_do210	------	------	------	------	------	------	------	------	------
MDMAIN_do2000	3990.1	3092.7	4167.9	612.94	811.45	857.42	390.46	710.84	216.45
NRMLKT_do300	------	------	------	------	------	------	------	------	------
POTENG_do1000	0.0417	0.0161	0.0154	0.0053	0.0084	0.0086	0.0086	0.0086	0.0086
POTENG_do2000	27.221	22.116	35.251	4.3611	5.4817	5.3145	1.3333	5.3315	1.3307
PREDIC_do1000	0.2692	0.4178	0.4178	0.4272	0.6795	0.6795	0.6800	0.1325	0.0388
RAND_do20	0.0000	0.0002	0.0555	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
RAND_do50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

4.5 Optimization factors from total time

SUBROUTINE	fx8	fx8 +		cedar optimization									
	manual	global	GP	first					best				
				1-cl	4-cl	XO	XS	CS	1-cl	4-cl	XO	XS	CS
MDMAIN_do2000	612.9400	811.4500	(32.4%)	857.4200	390.4600	(5.7%/	2.20/	1.57)	710.8400	216.4500	(-12.4%/	3.28/	2.83)
INTERF_do1000	512.6200	662.4900	(29.2%)	683.8100	253.1800	(3.2%/	2.70/	2.02)	647.1000	169.6600	(-2.3%/	3.81/	3.02)
CORREC_do1000	8.4606	18.5390	(119.1%)	39.6140	39.3450	(113.7%/	1.01/	0.22)	2.2551	20.1360	(-87.8%/	0.11/	0.42)
POTENG_do2000	43.7100	55.0700	(26.0%)	53.3450	13.3840	(-3.1%/	3.99/	3.27)	53.5040	13.3690	(-2.8%/	4.00/	3.27)
PREDIC_do1000	44.0170	69.0910	(57.0%)	69.2300	69.0890	(0.2%/	1.00/	0.64)	13.7130	4.1596	(-80.2%/	3.30/	10.58)
INTRAF_do1000	8.4214	11.3930	(35.3%)	16.8310	16.7200	(47.7%/	1.01/	0.50)	1.914	1.584	(-84 %/	1.2 /	5.31)
BNDRY_do100	0.3526	0.8803	(149.7%)	0.8270	0.8701	(-6.1%/	0.95/	0.41)	0.8615	0.8509	(-2.1%/	1.01/	0.41)
INTRAF_do2000	0.1621	0.2378	(46.7%)	0.2375	0.2378	(-0.1%/	1.00/	0.68)	0.2349	0.2528	(-1.2%/	0.93/	0.64)
MDG_do101	0.2506	0.1617	(-35.5%)	0.2082	0.2098	(28.8%/	0.99/	1.19)	0.1783	0.1415	(10.3%/	1.26/	1.77)
KINETI_do100	0.0947	0.1290	(36.2%)	0.1289	0.1538	(-0.1%/	0.84/	0.62)	0.1295	0.1295	(0.4%/	1.00/	0.73)
INITIA_do400	0.0003	0.0005	(66.6%)	0.0437	0.1193	(8638.3%/	0.37/	0.00)	0.0530	0.1220	(104979%/	0.43/	0.00)
INITIA_do2000	0.0559	0.0903	(61.5%)	0.0997	0.1073	(10.4%/	0.93/	0.52)	0.1007	0.1072	(11.5%/	0.94/	0.52)
MDG_do100	0.0525	0.0000	(-100.0%)	0.0512	0.0520	(999990%/	0.98/	1.01)	0.0877	0.0885	(999999%/	0.99/	0.59)
POTENG_do1000	0.0610	0.0915	(50.0%)	0.1203	0.1181	(31.5%/	1.02/	0.52)	0.0862	0.0863	(-5.8%/	1.00/	0.71)
INITIA_do300	0.0461	0.0531	(15.2%)	0.0473	0.0507	(-10.9%/	0.93/	0.91)	0.0573	0.0591	(7.9%/	0.97/	0.78)
CNSTNT_do1000	0.0027	0.0643	(2281.4%)	0.0033	0.0029	(-94.9%/	1.14/	0.93)	0.0548	0.0546	(-14.8%/	1.00/	0.05)
INITIA_do200	0.0016	0.0020	(25.0%)	0.0236	0.0264	(1079.9%/	0.89/	0.06)	0.0311	0.0239	(1454.9%/	1.30/	0.07)
INITIA_do100	0.0048	0.0075	(56.2%)	0.0089	0.0089	(18.7%/	1.00/	0.54)	0.0089	0.0089	(18.7%/	1.00/	0.54)
RAND_do50	0.0142	0.0078	(-45.1%)	0.0078	0.0078	(-0.0%/	1.00/	1.82)	0.0078	0.0078	(-0.0%/	1.00/	1.82)
RAND_do20	0.0002	0.0002	(-0.0%)	0.0002	0.0002	(-0.0%/	1.00/	1.00)	0.0002	0.0002	(-0.0%/	1.00/	1.00)
CSHIFT_do100	------	------	(-----)	------	------	(----- /	--- /	---)	------	------	(----- /	--- /	---)
INITIA_do1000	------	------	(-----)	------	------	(----- /	--- /	---)	------	------	(----- /	--- /	---)
INTERF_do2000	0.1526	0.1939	(27.1%)	0.1900	0.1809	(-2.0%/	1.05/	0.84)	------	------	(-100.0%/	--- /	---)
MDG_do210	------	------	(-----)	------	------	(----- /	--- /	---)	------	------	(----- /	--- /	---)
NRMLKT_do300	------	------	(-----)	------	------	(----- /	--- /	---)	------	------	(----- /	--- /	---)

4.6 Optimization factors from Min times

```
SUBROUTINE       fx8     fx8 +           |                      cedar optimization
                manual   global     GP   |             first                                   best
                                         | 1-cl    4-cl        XO     XS   CS      1-cl    4-cl       XO     XS    CS
 MDMAIN_do2000 612.9400 811.4500 (  32.4%)857.4200 390.4600 (   5.7%/ 2.20/ 1.57)710.8400 216.4500 ( -12.4%/ 3.28/ 2.83)
 INTERF_do1000   5.0462   6.5213 (  29.2%)  6.7372   2.4555 (   3.3%/ 2.74/ 2.06)  6.3655   1.6510 (  -2.4%/ 3.86/ 3.06)
 POTENG_do2000   4.3611   5.4817 (  25.7%)  5.3145   1.3333 (  -3.1%/ 3.99/ 3.27)  5.3315   1.3307 (  -2.7%/ 4.01/ 3.28)
 PREDIC_do1000   0.4178   0.4272 (   2.2%)  0.6795   0.6795 (  59.1%/ 1.00/ 0.61)  0.6800   0.1325 (  59.2%/ 5.13/ 3.15)
 PREDIC_do1000   0.4272   0.6795 (  59.1%)  0.6795   0.6800 (  -0.0%/ 1.00/ 0.63)  0.1325   0.0388 ( -80.5%/ 3.41/11.01)
 PREDIC_do1000   0.4272   0.6795 (  59.1%)  0.6795   0.6800 (  -0.0%/ 1.00/ 0.63)  0.1152   0.1119 ( -83.0%/ 1.03/ 3.82)
     MDG_do101   0.2506   0.1617 ( -35.5%)  0.2082   0.2098 (  28.8%/ 0.99/ 1.19)  0.1783   0.1415 (  10.3%/ 1.26/ 1.77)
 CORREC_do1000   0.0821   0.1795 ( 118.6%)  0.3818   0.3818 ( 112.7%/ 1.00/ 0.22)  0.0217   0.0740 ( -87.9%/ 0.29/ 1.11)
 INTRAF_do1000   0.0811   0.1098 (  35.4%)  0.1610   0.1610 (  46.6%/ 1.00/ 0.50)  0.0178   0.0086 ( -84  %/ 2.06/ 9.43)
 INITIA_do2000   0.0559   0.0903 (  61.5%)  0.0997   0.1073 (  10.4%/ 0.93/ 0.52)  0.1007   0.1072 (  11.5%/ 0.94/ 0.52)
     MDG_do100   0.0525   0.0000 (-100.0%)  0.0512   0.0520 (999999%/ 0.98/ 1.01)  0.0877   0.0885 (999999%/ 0.99/ 0.59)
  INITIA_do300   0.0461   0.0531 (  15.2%)  0.0473   0.0507 ( -10.9%/ 0.93/ 0.91)  0.0573   0.0591 (   7.9%/ 0.97/ 0.78)
 POTENG_do1000   0.0053   0.0084 (  58.5%)  0.0086   0.0086 (   2.4%/ 1.00/ 0.62)  0.0086   0.0086 (   2.4%/ 1.00/ 0.62)
  INITIA_do100   0.0048   0.0075 (  56.2%)  0.0089   0.0089 (  18.7%/ 1.00/ 0.54)  0.0089   0.0089 (  18.7%/ 1.00/ 0.54)
   BNDRY_do100   0.0035   0.0082 ( 134.3%)  0.0082   0.0082 (  -0.0%/ 1.00/ 0.43)  0.0082   0.0082 (  -0.0%/ 1.00/ 0.43)
 CNSTNT_do1000   0.0027   0.0643 (2281.4%)  0.0033   0.0029 ( -94.9%/ 1.14/ 0.93)  0.0548   0.0546 ( -14.8%/ 1.00/ 0.05)
  INITIA_do200   0.0016   0.0020 (  25.0%)  0.0236   0.0264 (1079.9%/ 0.89/ 0.06)  0.0311   0.0239 (1454.9%/ 1.30/ 0.07)
 INTRAF_do2000   0.0015   0.0023 (  53.3%)  0.0023   0.0023 (  -0.0%/ 1.00/ 0.65)  0.0023   0.0023 (  -0.0%/ 1.00/ 0.65)
 INTERF_do2000   0.0013   0.0017 (  30.8%)  0.0017   0.0017 (  -0.0%/ 1.00/ 0.76)  ------   ------ (-100.0%/ --- / --- )
  KINETI_do100   0.0009   0.0012 (  33.3%)  0.0012   0.0012 (  -0.0%/ 1.00/ 0.75)  0.0012   0.0012 (  -0.0%/ 1.00/ 0.75)
  INITIA_do400   0.0003   0.0005 (  66.6%)  0.0437   0.1193 (8638.3%/ 0.37/ 0.00)  0.0530   0.1220 (104979%/ 0.43/ 0.00)
     RAND_do20   0.0002   0.0002 (  -0.0%)  0.0002   0.0002 (  -0.0%/ 1.00/ 1.00)  0.0002   0.0002 (  -0.0%/ 1.00/ 1.00)
  CSHIFT_do100   ------   ------ ( ----- )  ------   ------ ( ----- / --- / --- )  ------   ------ ( ----- / --- / --- )
 INITIA_do1000   ------   ------ ( ----- )  ------   ------ ( ----- / --- / --- )  ------   ------ ( ----- / --- / --- )
     MDG_do210   ------   ------ ( ----- )  ------   ------ ( ----- / --- / --- )  ------   ------ ( ----- / --- / --- )
  NRMLKT_do300   ------   ------ ( ----- )  ------   ------ ( ----- / --- / --- )  ------   ------ ( ----- / --- / --- )
     RAND_do50   0.0000   0.0000 (-100.0%)  0.0000   0.0000 (-100.0%/ 0.00/ 0.00)  0.0000   0.0000 (-100.0%/ 0.00/ 0.00)
```

4.7 Profiles with the static scheduling runtime library

4 cluster execution. Mimimum, average, and total times. Alphabetically sorted. (loop intraf/1000 is a different version. Not comparable to timing results above)

```
                   minimum     average       total

      MDG_do100    0.043800    0.043800      0.043800
      MDG_do100    0.043800    0.043800      0.043800
      MDG_do101    0.094760    0.094760      0.094760
      MDG_do101    0.094760    0.094760      0.094760
      RAND_do20    0.000220    0.000220      0.000220
      RAND_do50    0.000050    0.000059      0.007550
    BNDRY_do100    0.008280    0.008929      0.892900
   INITIA_do100    0.007860    0.007860      0.007860
   INITIA_do200    0.033160    0.033160      0.033160
   INITIA_do300    0.052240    0.052240      0.052240
   INITIA_do400    0.133790    0.133790      0.133790
   KINETI_do100    0.001280    0.001667      0.166690
  CNSTNT_do1000    0.053560    0.053560      0.053560
  CORREC_do1000    0.092470    0.104885     10.488530
  INITIA_do2000    0.115250    0.115250      0.115250
  INTERF_do1000    2.205870    2.241083    226.349390
  INTRAF_do1000    0.116190    0.120344     12.154750
  INTRAF_do2000    0.002370    0.002375      0.239900
  MDMAIN_do2000  283.652280  283.652280    283.652280
  POTENG_do1000    0.008630    0.008633      0.086330
  POTENG_do2000    2.411130    2.424761     24.247610
  PREDIC_do1000    0.110100    0.115954     11.595390
```