Funded by the European Union

Optimizing physics subroutines for multithreaded processors

ESCAPE

• What is ESCAPE?

ESCAPE is a H2020 Project about Energy efficient and SCalable Algorithms for weather Predction at Exascale

• What is the goal?

The goal is to demonstrate improvements in performance for a selected set of algorithms referred to as "**dwarfs**".

DMI works on a dwarf version of the ACRANEB2 radiation scheme, which has been optimized for running on multithreaded processors.

• Why?

The current 1D physics schemes are embarrassingly scalable. Still we show that they can be made much more efficient than at present.

• But is not OpenMP implemented already?

Only coarse-grained OpenMP has been implemented in HARMONIE, which is far from optimal (O'Brien, All Staff Workshop, 2015). Quote from the presentation:

"Performance is poor partly because HARMONIE runs out of memory before running out of threads! Changes should be made to avoid memory blow-up!"

• How can it be done?

1. Original loop:

```
DO JLEV=KTDIA,KLEV

  ! compute unscaled absorber amounts 2.du
  DO JLON=KIDIA,KFDIA
    ZDU(JLON,1)=2._JPRB*PDELP(JLON,JLEV)*ZQ   (JLON,JLEV)
    ZDU(JLON,2)=2._JPRB*PDELP(JLON,JLEV)*PQCO2(JLON,JLEV)* &
     &          (1._JPRB-ZQ(JLON,JLEV))
    ZDU(JLON,3)=2._JPRB*PDELP(JLON,JLEV)*PQO3 (JLON,JLEV)* &
     &          (1._JPRB-ZQ(JLON,JLEV))
    ZDU(JLON,4)=ZDU(JLON,1)*RV*ZQ(JLON,JLEV)/(RD+(RV-RD)*ZQ(JLON,JLEV))
  ENDDO

  ! compute total and incremental optical depths
  CALL DELTA_C(JLEV,PAPRSF(1,JLEV),PT(1,JLEV),ZDU,   &
   & ZC_UW,ZC_US,ZC_US_IRHOV,ZC_UC,ZC_U,ZC_PU,ZC_TU, &
   & ZDEOTA0(1,JLEV))
  CALL DELTA_T(JLEV,PAPRSF(1,JLEV),PT(1,JLEV),ZDU,   &
   & ZT_UW,ZT_US,ZT_US_IRHOV,ZT_UC,ZT_U,ZT_PU,ZT_TU, &
   & ZDEOTA1(1,JLEV))
  DO JLON=KIDIA,KFDIA
    PDEOTI(JLON,JLEV)=MAX(ZDEOTA1(JLON,JLEV)-ZDEOTA1(JLON,JLEV-1),0._JPRB)
  ENDDO

ENDDO
```

2. Switched loop order and reduced array sizes:

```
DO JLEV=KTDIA,KLEV

  ! compute unscaled absorber amounts 2.du
  ZDU(1)=2._JPRB*PDELP(JLEV,JLON)*ZQ   (JLEV,JLON)
  ZDU(2)=2._JPRB*PDELP(JLEV,JLON)*PQCO2(JLEV,JLON)* &
   &          (1._JPRB-ZQ(JLEV,JLON))
  ZDU(3)=2._JPRB*PDELP(JLEV,JLON)*PQO3 (JLEV,JLON)* &
   &          (1._JPRB-ZQ(JLEV,JLON))
  ZDU(4)=ZDU(1)*RV*ZQ(JLEV,JLON)/(RD+(RV-RD)*ZQ(JLEV,JLON))
  ZP_SCA=PAPRSF(JLEV,JLON)
  ZT_SCA=PT(JLEV,JLON)

  ! compute total and incremental optical depths
  CALL DELTA_C_SCA(JLON,JLEV,ZP_SCA,ZT_SCA,ZDU,ZC_UW,ZC_US, &
   & ZC_US_IRHOV,ZC_UC,ZC_U,ZC_PU,ZC_TU,ZDEL0)
  CALL DELTA_T_SCA(JLON,JLEV,ZP_SCA,ZT_SCA,ZDU,ZT_UW,ZT_US, &
   & ZT_US_IRHOV,ZT_UC,ZT_U,ZT_PU,ZT_TU,ZDEL1)
  ZDEOTA0(JLEV,JLON)=ZDEL0
  ZDEOTA1(JLEV,JLON)=ZDEL1
  PDEOTI(JLEV,JLON)=MAX(ZDEOTA1(JLEV,JLON)-ZDEOTA1(JLEV-1,JLON),0._JPRB)
ENDDO
```

3. As above but scalar variables used as far as possible:

```
do jlev2=jlev1+1,ilev  ! final half level

  ! compute unscaled absorber amounts 2.du
  zdu1=two*pdelp(jlev2,jlon)*zq   (jlev2)
  zdu2=two*pdelp(jlev2,jlon)*pqco2(jlev2,jlon)*(one-zq(jlev2))
  zdu3=two*pdelp(jlev2,jlon)*pqo3 (jlev2,jlon)*(one-zq(jlev2))
  zdu4=zdu1*rv*zq(jlev2)/(rd+(rv-rd)*zq(jlev2))
  zp_sca=paprsf(jlev2,jlon)
  zt_sca=pt(jlev2,jlon)

  ! compute optical depths
  zc_uc = zc_uc+zc_fc(jlev2)*zdu4*zmd
  jg=1
  zc_uw1 = zc_uw1+zc_fw(jlev2,jg)*zdu1
  zc_us1 = zc_us1+zc_fs(jlev2,jg)*zdu1*zmd
  zc_us_irhov1 = zc_us_irhov1+zirhov(jlev2)*zc_fs(jlev2,jg)*zdu1*zmd
  ! update u and t_avg.u
  zc_ul  = zc_ul+        zdu1
  zc_pul = zc_pul+zp_sca*zdu1
  zc_tul = zc_tul+zt_sca*zdu1
  call delta_c_sca_scalar1(zdu1,zc_uw1,zc_us1,zc_us_irhov1,zc_uc,     &
                           zc_ul,zc_pul,zc_tul,zdelta1)
```

As it can be seen, this is not complicated, nor is the new code unreadable.

ACRANEB2 has more than 15 500 lines. The other physics routines have less. Improvements as these are feasible implement throughout these!

Relative physics time	Lines (approx.)	subroutine	Function
22.6%	4300	ARO_RAIN_ICE	Precipitation microphysics
17.4%	3800	VDFHGHTHL	Shallow mass flux mixing
13.0%	5700	ARO_TURB_MNH	Turbulence sources and TKE evolution
12.2%	5800	ARO_SHALLOW_MF	Turbulence mass flux and divergence
5.9%	1500	ARO_ADJUST	Adjustment of cloud liquid and frozen water
4.4%	3100	APL_AROME	Main physics routine (local lines only)
Sum = 75.51%	Sum = 24200	6 of 17 subroutines	

2017-04-04

The DMI ESCAPE team: B. H. Sass, J. W. Poulsen, K. P. Nielsen and P. Berg

ALADIN Workshop & HIRLAM All Staff Meeting 2017, Helsinki, Finland