3.2 Precision Switch Conversion

One feature of ECHAM radiation, contained in the module `rk_mo_kind`, is a kind parameter `wp` that defines the precision of all `REAL(wp)` variables in the code.

![Diagram showing code structure change]

**Figure 2.** Code structure change, modifications in red.

```fortran
    INTEGER, PARAMETER :: sp = SELECTED_REAL_KIND(ps,rs)
    INTEGER, PARAMETER :: dp = SELECTED_REAL_KIND(pd,rd)
    [...]  
    INTEGER, PARAMETER :: wp = dp
```

A such precision switch would be a really elegant way to change the precision of ECHAM. This feature is unfortunately not fully implemented through the original code, since `dp` is used instead in several Fortran modules. By assigning 4 to the value of `dp` and declaring an additional precision "hp" for actual double precision where needed, the previous problem can be circumvented. In this manner, compilation was possible for the program after some modifications (MPI, NetCDF, mo_echam_radkernel_cross_messages). The compiled code was crashing at runtime because of internal ECHAM bugs triggered by code in `rk_mo_srtm_solver` and other parts (these issues were solved later when investigating each code part with the incremental conversion method). The root cause of these bugs could not be easily tracked in many cases.

3.3 Code Block Conversion

Another test consisted in the precision conversion of time-consuming code subroutines and functions. This required further modifications in the code, such as conversion of input and output variables to comply with Fortran syntax:

```fortran
    This procedure allowed effective implementation of single-precision, showing AAAAAAAAAAAAA40% peak performance gain in some code parts. At the same time the conversion overhead was generally detrimental for the model time performance. This overhead issue could be solved by applying the scheme showed in picture 2 to a bigger code-block, so that the variables converted are few in relation to the block time consumption. Radiation was a perfect candidate for this purpose.
```