COMPILERS AND PROBLEMS WITH CMAQ 5.1 TO 5.2.1
George Delic, Ph.D., HiPERISIM Consulting, LLC

Introduction

This presentation reports on problems encountered while implementing the parallel sparse matrix solver, FSparse [1], in the Chemistry Transport Model (CTM) in CMAQ. In this report issues encountered with both the original EPA [2], and FSparse, algorithms are listed for the GEAR version of the CTM.

Test bed environment

The hardware systems chosen were the platforms at HiPERISIM Consulting, LLC, consisting of two nodes with dual 16-core Intel E5v3 CPUs on each node. These are the base nodes of a heterogeneous cluster that includes an HP blade server hosting eight nodes with dual 4-core Intel E5640 CPUs. The total core count of this cluster is 128. The MPI executions are launched across multiple combination of these nodes. This cluster allows for comparison of the FSparse hybrid (MPI + OpenMP) parallel versions of CMAQ with the original EPA version (MPI only).

Compilers

Issues uncovered and reported here involved both the Intel Parallel Studio® suite (releases 17.2 and 17.6) and Portland Group compiler (release 18.1) with compiler options for a heterogeneous cluster. In the Portland case, utilization of an Intel wrapper enabled linking to the Intel MPI library. Tracking down issues was often a long and laborious process. Observations are presented as unordered lists taken from build or run logs for CMAQ 5.1.2, and 5.2.1.

Critical compiler switches

Some observations on compiler flag choices to avoid runtime errors:
- use of -init-arrays,zero to initialize arrays to zero (CMAQ has uninitialized arrays)
- use of -axSE4.2 to enable a executable to run on a heterogeneous cluster
- use of -02 to avoid runtime errors when using -O3
- use of -ip-no-inlining to avoid runtime errors due in part to using -ipo
- use of -warn all,declarations,nounused (to avoid compilation termination)
- replacement of “ifdef parallel” with “ifdef parallel_mpi”

Mixed mode arithmetic

While some progress to fix this issue has been made in recent releases of the EPA version of CMAQ, mixed mode arithmetic continues to occur in CMAQ. This is due to use of single precision (SP) floating point operations (FP) throughout CMAQ except for the CTM where double precision (DP) is used. Thus, of necessity, SP variables are passed into the CTM and this leads to issues such as follows:
- Inconsistent conversion to DP that leads to random digits in the second half of FP words.
- Expressions that mix SP and DP variables (or constants).
- Constants that are not defined as DP (e.g. 1.0d0 and not 1.0)
- Constants that have limited precision even in SP
Where ever these issues were detected, they were corrected in the FSparse version of the CTM but they remain in the Original U.S. EPA release.

Fortran Lint summary analysis

While not all library interfaces are included (e.g. IOAPI, NETCDF, MPI) this is the summary statistics report for CMAQ 5.2.1 from Fortran Lint [3] listing potential problem areas of the source code.

<table>
<thead>
<tr>
<th>Fortran lint summary analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fortran lint Rev: 6.02</td>
</tr>
<tr>
<td>Date: 31-Oct-2018</td>
</tr>
<tr>
<td>Source files: 571</td>
</tr>
<tr>
<td>Include files: 85100 lines</td>
</tr>
<tr>
<td>Total compiled: 121910 lines</td>
</tr>
<tr>
<td>Subroutines: 164</td>
</tr>
<tr>
<td>Functions: 61</td>
</tr>
<tr>
<td>Error count: 1697</td>
</tr>
<tr>
<td>Statistics:</td>
</tr>
<tr>
<td>Number of source files: 571</td>
</tr>
<tr>
<td>Include files: 85100 lines</td>
</tr>
<tr>
<td>Total compiled: 121910 lines</td>
</tr>
<tr>
<td>Subroutines: 164</td>
</tr>
<tr>
<td>Functions: 61</td>
</tr>
<tr>
<td>Error count: 1697</td>
</tr>
<tr>
<td>Statistics:</td>
</tr>
<tr>
<td>Number of source files: 571</td>
</tr>
<tr>
<td>Include files: 85100 lines</td>
</tr>
<tr>
<td>Total compiled: 121910 lines</td>
</tr>
<tr>
<td>Subroutines: 164</td>
</tr>
<tr>
<td>Functions: 61</td>
</tr>
<tr>
<td>Error count: 1697</td>
</tr>
</tbody>
</table>

Examples of uncovered issues

1. ifort bug: FALSE values produced with -O2 or -O3 for both FSparse or JSparse
2. No type declaration for SETLAM (in lat Ion.F) in ICON and BCON
3. RUNTIME: ERROR ABORT in subroutine READET in readet.f
4. compile time bug in the interface between these two files
   - /models/JPROC/jproc_table/abrands.f
   - /models/JPROC/jproc_table/iatvats.f
5. ipo: warning #11012: unresolved nclos
6. lack of type definitions; many warnings of the type:
   - /srispa.inc(38): warning #6717: This name has not been given an explicit type PARAMETER
     (NCOMP=8,NIONS=10,NGASAQ=3,NSLDS=19,NPAIR=23,NZS R=100,
7. Global name too long
8. 12 messages of the type (compiler removed "se_" from name):
9. warning #5194: Source line truncated
10. 19 messages of the type:
    - LSM_MOD.F(143): warning #5194: Source line truncated
11. 8 messages of the type:
    - se_data_send_module.f(351): warning #6843: A dummy argument with an explicit INTENT(OUT) declaration is not given an explicit value. [REQUEST]
12. preprocessing of MPI parallelism in CMAQ: replace 89 occurrences of
    - set PAR = (-Dparallel), by
    - set PAR = (-Dparallel_mpi)
13. error in C/lcm/src/emis/ems/LTNG_DEFN.FLTNG_DEFN.F(301):
    error #640. This name does not have a type, and must have an explicit type. [SETLAM]
14. NaVs at runtime in../aero/aero6/AEROSOL_chemistry.F
15. 8 messages of the type:
    - se_data_send_module.f(351): warning #6843: A dummy argument with an explicit INTENT(OUT) declaration is not given an explicit value. [REQUEST]

Conclusions

This report has described an analysis of CMAQ source code across three releases in the standard U.S. EPA model and shows problems in each case. While some problems are related to compiler versions (and can vary between them), most are in the source code itself. This suggest more careful code validation steps are appropriate using any available code analysis software.

References

[3] For a detailed list of issues found, or for information on compilers and Fortran Lint analysis software, contact george@hiperism.com