Karlsruher Institut für Technologie

C10-Assessment of COBRA-TF source at KIT

J. Jiménez and V. Sanchez

Institute for Neutron Physics and Reactor Technology (INR)
Reactor Physics and Dynamic Group (RPD)

Content

- Results of the non-regression test
- Assessment of the source code at KIT
  - Two internal reports submitted to the maintenance team
    - Bugs and errors found in the source code of the version received on November 2013 (21.11.2013)
    - Fixing the restart capabilities of the version received on November 2013 (23.03.2014)
- Debugging tools, what to do when you get a segmentation fault signal
- Conclusions

Results of the non-regression test

- Those results correspond to the version received on 11.04.2014
- Intel 14.0.1 with RELEASE compilation options
  - 54% tests passed, 101 tests failed out of 221
- Intel 14.0.1 with DEBUG compilation options
  - 58% tests passed, 92 tests failed out of 221

Weak portability of the code

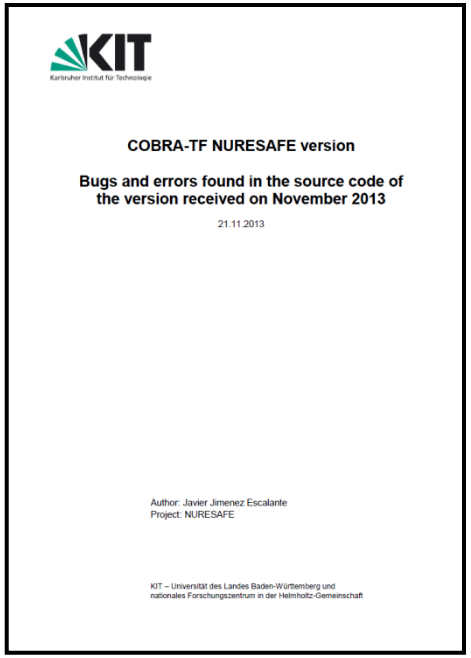
- GNU 4.8.2 with RELEASE compilation options
  - 99% tests passed, 1 tests failed out of 221
- GNU 4.8.2 with DEBUG compilation options
  - 96% tests passed, 9 tests failed out of 221

With Debugging options things are also not so good

The same compiler produces not-running code in some cases, that is a big problem

Assessment of the source code at KIT

- Two reports submitted to the maintenance team. Effective communication via email exchange.



Karlsruher Institut für Technologie

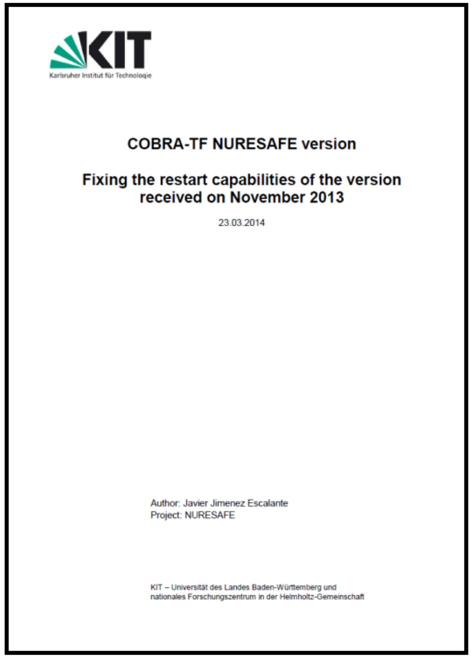
COBRA-TF NURESAFE version

Bugs and errors found in the source code of the version received on November 2013

21.11.2013

Author: Javier Jimenez Escalante
Project: NURESAFE

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft



Karlsruher Institut für Technologie

COBRA-TF NURESAFE version

Fixing the restart capabilities of the version received on November 2013

23.03.2014

Author: Javier Jimenez Escalante
Project: NURESAFE

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

- The restart capabilities of the CTF version doesn´t properly
  - The new modules have not been included in dumpit.f and restart.f. (rks171)
  - Modules parallel, solid_material_props, transfer_io, vtk, powermod
- When reading the restart, variables are not allocated and code crashes

What to do when you get a segmentation fault

```
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image              PC                Routine            Line        Source
libintlc.so.5      00007F71BB2E6229  Unknown               Unknown  Unknown
libintlc.so.5      00007F71BB2E4BA0  Unknown               Unknown  Unknown
libifcore.so.5     00007F71BC9C333F  Unknown               Unknown  Unknown
libifcore.so.5     00007F71BC99AD7E  Unknown               Unknown  Unknown
libi                                                                    own
libi                                                                    own
cobr                                                                    own
cobr                                                                    own
cobr                                                                    own
cobratf            00000000004078A1  Unknown               Unknown  Unknown
cobratf            0000000000404067  Unknown               Unknown  Unknown
cobratf            0000000000401CA8  Unknown               Unknown  Unknown
cobratf            0000000000401BF6  Unknown               Unknown  Unknown
libc.so.6          00007F71BAD29C85  Unknown               Unknown  Unknown
cobratf            0000000000401AD9  Unknown               Unknown  Unknown
```

A *segmentation fault* occurs when a program attempts to **access a memory location that it is not allowed to access**, or attempts to access a memory location **in a way that is not allowed** (for example, attempting to write to a **read-only location**, or to overwrite part of the operating system)

Go and fix your code

But, Where to look?

Running the test cases the code crashes

```
[jimenez@inrsim06 MYBUILD_GFOR_DEBUG]$ make test
Running tests...
Test project /home/jimenez/COBRA-TF-master/MYBUILD_GFOR_DEBUG
        Start   1: COBRA_TFUtils_CTF_VTK_Diff_Tool
  1/221 Test   #1: COBRA_TFUtils_CTF_VTK_Diff_Tool ...................   Passed    1.41 sec
        Start   2: COBRA_TFUtils_CTF_EXACT_DIFF_TOOL_TEST
  2/221 Test   #2: COBRA_TFUtils_CTF_EXACT_DIFF_TOOL_TEST ............   Passed    0.06 sec
        Start   3: COBRA_TFUtils_par_file_copy_test
  3/221 Test   #3: COBRA_TFUtils_par_file_copy_test ..................   Passed    0.10 sec
        Start   4: COBRA_TFUtils_par_file_diff_test
  4/221 Test   #4: COBRA_TFUtils_par_file_diff_test ..................   Passed    0.17 sec
        Start   5: COBRA_TFUtils_test_the_unit_test_harness
  5/221 Test   #5: COBRA_TFUtils_test_the_unit_test_harness ..........   Passed    0.01 sec
        Start   6: COBRA_TFCore_ctf_unit_tests
  6/221 Test   #6: COBRA_TFCore_ctf_unit_tests .......................   Passed    0.93 sec
        Start   7: COBRA_TFPreproc_preproc_unit_tests
  7/221 Test   #7: COBRA_TFPreproc_preproc_unit_tests ................   Passed    0.02 sec
        Start   8: COBRA_TF_preproc_small_singlerod_HFP
  8/221 Test   #8: COBRA_TF_preproc_small_singlerod_HFP ..............   Passed    0.07 sec
        Start   9: COBRA_TF_run_small_singlerod_HFP
*** Error in `/home/jimenez/COBRA-TF-master/MYBUILD_GFOR_DEBUG/cobra_tf/test_matrix/../ctf_src/cobratf':
free(): invalid pointer: 0x00000000019b4a10 ***
*** Error in `/home/jimenez/COBRA-TF-master/MYBUILD_GFOR_DEBUG/cobra_tf/test_matrix/../ctf_src/cobratf':
malloc(): memory corruption: 0x00000000019b4a40 ***
```

We get a weird error, try to isolate it and make it reproducible

Then go to the folder and do a local execution

```
[jimenez@inrsim06 COBRA_TF_run_small_3x3rod_HFP]$ ../../../cobra_tf/ctf_src/cobratf
Reading CARD GROUP     1
Reading CARD GROUP     2
Reading CARD GROUP     3
Reading CARD GROUP     4
Reading CARD GROUP     7
Reading CARD GROUP     8
Reading CARD GROUP     9
Reading CARD GROUP    10
Reading CARD GROUP    11
Reading CARD GROUP    12
Reading CARD GROUP    13
Reading CARD GROUP    14
Reading CARD GROUP    15
Reading CARD GROUP    17
Reading CARD GROUP    18
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image              PC                Routine            Line        Source
libintlc.so.5      00007EEFC57F6229  Unknown               Unknown  Unknown
libintlc.so.5      00007EEFC57F4BA0  Unknown               Unknown  Unknown
libifcore.so.5     00007EEFC6ED333F  Unknown               Unknown  Unknown
libifcore.so.5     00007EEFC6E3AD7F  Unknown               Unknown  Unknown
libifcore.so.5     00007EEFC6E4BF83  Unknown               Unknown  Unknown
libpthread.so.0    00007EEFC55DC760  Unknown               Unknown  Unknown
cobratf            00000000006DB240  Unknown               Unknown  Unknown
cobratf            00000000006DE152  Unknown               Unknown  Unknown
```

We verify that this input deck crashes the code

Try again using DEBUG and gdb debugger

```
[jimenez@inrsim06 COBRA_TF_run_small_3x3rod_HFP]$ gdb ../../../cobra_tf/ctf_src/cobratf
GNU gdb (GDB) 7.6-6.mga4 (Mageia release 4)
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-mageia-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/jimenez/COBRA-TF-master/MYBUILD_IFORT_DBG/cobra_tf/ctf_src/cobratf...done.
(gdb) run
Starting program: /home/jimenez/COBRA-TF-master/MYBUILD_IFORT_DBG/cobra_tf/ctf_src/cobratf
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib64/libthread_db.so.1".
Reading CARD GROUP     1
Reading CARD GROUP     2
Reading CARD GROUP     3
Reading CARD GROUP     4
Reading CARD GROUP    ...
Reading CARD GROUP    15
Reading CARD GROUP    17
Reading CARD GROUP    18
Program received signal SIGSEGV, Segmentation fault.
0x00000000006db240 in MOD_GAUSS::gauss (n=5, ain=..., bin=..., cin=..., ddin=..., tt1=...) at
/home/jimenez/COBRA-TF-master/cobra_tf/ctf_src/Mod_gauss.f90:51
51              s(1) = dd(1)
```

Source file and the line

Find where is the problem and why it happens

```
[jimenez@inrsim06 COBRA_TF_run_small_3x3rod_HFP]$ vi ~/COBRA-TF-master/cobra_tf/ctf_src/Mod_gauss.f90

subroutine gauss(n,ain,bin,cin,ddin,tt1)
      implicit none

      integer, intent(in) :: n
      real   , intent(in) :: ain(:)
      real   , intent(in) :: bin(:)
      real   , intent(in) :: cin(:)
      real   , intent(in) :: ddin(:)
      real   , intent(out) :: tt1(:)

      real, allocatable :: a(:)
      real, allocatable :: b(:)
      real, allocatable :: c(:)
      real, allocatable :: dd(:)
      real :: s(n)

      ! Set the data arrays passed in to the local working arrays
      allocate (a(size(ain)),b(size(bin)),c(size(cin)),dd(size(ddin)))
      a = ain
      b = bin
      c = cin
      dd = ddin
```

Input parameters of unknown size

Local variables were we want to operate without modifying the original ones

Copying of arrays on something that still is not allocated produce errors !!! COMPILER DEPENDENT!!!

Repeat the operation till the code runs properly

```
 Cross check of the input values finished.
 Rank            0  Step          100 Energy Storage =    74.7980456802663
 Rank            0  Step          200 Energy Storage =    33.0864953124636
 Rank            0  Step          300 Energy Storage =    4.31131860702695
Convergence Reached, Ending Pseudo Transient
Number of Steps =   386
         Quantity  Value (%)  Criteria (%)
 Global Energy Balance =     -0.0003        0.0100
   Global Mass Balance =     -0.0100        0.0100
  Fluid Energy Storage =      0.0023        0.5000
          Mass Storage =      0.0100        0.5000
  Solid Energy Storage =      0.0531        0.5000

Program received signal SIGSEGV, Segmentation fault.
0x0000000000588fd3 in MOD_VTK_LEGACY::mdot_cell_center (mdot_cell_center=..., geodat=..., mdot=...,
alpha=..., rho=..., area_mom=..., area_cont=...)
    at /home/jimenez/COBRA-TF-master/cobra_tf/ctf_src/Mod_vtk_legacy.F90:1117
1117                rho_alpha_A=rho_alpha_A * area_mom(chan,level)
```

General tips

- Use several compilers (GNU, INTEL, etc)
- Try different sets of flags (debug and optimization)
- Use **ddd** (Graphical interface of gdb)
- **Valgrind** is also a powerful memory checker.

Valgrind output

```
==4303== More than 10000000 total errors detected.  I'm not reporting any more.
==4303== Final error counts will be inaccurate.  Go fix your program!
==4303== Rerun with --error-limit=no to disable this cutoff.  Note
==4303== that errors may occur in your program without prior warning from
==4303== Valgrind, because errors are no longer being displayed.
```

to find it is by doing printing.

- Find a small and fast running case which reproduce the bug.
- Introduce printing to track the code execution.

Conclusions

- The current situation is better than the previous one
- The development environment is fine (cmake, unit test, coverage reports)
- Nevertheless, the portability should be enhance:
  - Different compilers need to be used.
  - Incompatibility with gnu 4.4.3 in Mandriva 2010.2. Coding of Mod_model_maps.f90
- One need to know what he is doing:

```
Error: Unclassifiable statement at (1)
Error: Syntax error in argument list at (1)
Error: Syntax error in DO statement at (1)
Error: Expected another dimension in array declaration at (1)
Error: Invalid form of array reference at (1)
```