GROMACS - Bug #1934

QMMM with ORCA: memory leaks, buffer overflows and much more

03/31/2016 05:46 PM - Grzegorz Wieczorek

Status:	New		
Priority:	Normal		
Assignee:			
Category:			
Target version:			
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:	5.1.2		

Description

Dear GMXers!
The most important part regards the "much more" from the title. To warm up lets see the memory issues first.
qm_orca.c is abundant in memory leaks - allocations (some of them completely unnecessary) not followed by freeing the memory. It happens during initialization of QM software and every time QMMM is called. It is coupled with really bad environment variables handling that can lead to crashes and execution of arbitrary code, which can be dangerous if gmx is setuid/setgid (why should it be??). Few commented excerpts from qm_orca.c:

```
63:    char buf;
64:    snew(buf, 200); // OK, we got some memory
65:
66:    / ORCA settings on the system */
67:    buf = getenv("GMX_QM_ORCA_BASENAME"); // heh... the previous mem referenced by buf got leaked
68:    if (buf)
69:    {
70:        snew(qm->orca_basename, 200);
71:        sscanf(buf, "%s", qm->orca_basename); // buf points to the value of the env variable. It can be longer than 200 though...
```

Lets try the script1:

```
#!/bin/tcsh
setenv GMX_QM_ORCA_BASENAME `seq -s "" 1000`
setenv setenv GMX_ORCA_PATH /home/soft/orca_3_0_3_linux_x86-64

gmx_d grompp -f pyp.mdp -p pyp.top -n pyp.ndx -c pyp.gro -o pyp.tpr
#gmx_d compiled with orca, pyp.ndx, .gro and .top taken from http://wwwuser.gwdg.de/~ggroenh/qmmm.html#ex, pyp.mdp taken from thereof, modified, attached

gmx_d mdrun -ntmpi 1 -ntomp 1 -s pyp.tpr
```

Lets run:

```
% ./script1
(...)
Using 1 MPI thread
Using 1 OpenMP thread

QM/MM calculation requested.
Layer 0
nr of QM atoms 22
QMlevel: DFT/3-21G

Setting ORCA path to: /home/soft/orca_3_0_3_linux_x86-64...
ORCA initialised...

Segmentation fault (core dumped)
%
```

If GMX_QM_ORCA_BASENAME is set to something reasonable, it runs flawlessly, IF you remove %LJCoefficients and %pointcharges directives from the attached example.ORCAINFO.
Here comes the "much more". If requested by bOpt = true option in the .mdp, gmx does not create .pc nor .LJ files. 5 minutes of

debugging shows that it happens, because in qm_orca.c, write_orca_input:

183: if (QMMMrec->QMMMscheme != eQMMMschemeoniom && mm->nrMMatoms)

mm->nrMMatoms is 0. Going back along the execution path, init_QMMMrec in qmmm.c:

732: mm->nrMMatoms   = (mtop->natoms)-(qr->qm[0]->nrQMatoms); /* rest of the atoms */

which for this system yields with 15806 - thats OK. However, before even calling the qmmm calculation, in update_QMMMrec, qmmm.c, for serial execution:

```
791:  (...) mm_nr = 0 (...)
 851: if (QMMMlist.nri) // QMMMlist.nri is zero, so
 889:   mm_nr++;        // we don't get here, so ...
```

1012: mm->nrMMatoms = mm_nr; // number of MM atoms gets zeroed

My question is:
If you have some suggestions how to fix this, please let me know - as I really want it to work, I will gladly contribute. From the comments in the source I got that there is some work to be done in neighbor searching, but I don't know the details (yet). Discussion on gmx-users and gmx-developers on qmmm are very rare and not giving me too much of enlightment. Any input/guidance on this would be very welcome!
Best Regards,

Grzegorz Wieczorek

Related issues:

Related to GROMACS - Task #2706: Rework classic QM/MM interface	**Accepted**

History

#1 - 04/01/2016 03:24 AM - Grzegorz Wieczorek

- File qm_orca.c added

OK, in the meantime I fixed the leaks, the possibilities of overflow, reduced the number of unnecessary variables, simplified the way ORCA outputs are read and few other things. The patch is longer than the file itself, so I upload the file. I am sorry - I changed the indentation a bit - to more "conventional" - I wanted to see more code on the screen at a time. I checked, it seems to work not worse than the older version.

g

#2 - 04/01/2016 03:42 AM - Grzegorz Wieczorek

- File qm_orca.c added

Ehh, this is the latest.

#3 - 04/01/2016 08:55 PM - Grzegorz Wieczorek

- File qm_orca.c added

I've just read about your preferred code formatting and indentation standards, so I applied the new knowledge to the qm_orca.c (making it hardly readable again ;)). Here it is.

#4 - 04/04/2016 10:56 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1934.
Uploader: Grzegorz Wieczorek (gigo@ibb.waw.pl)
Change-Id: I999b7112b46b563d2b25fbaab8da4bcefb7d6d0f
Gerrit URL: https://gerrit.gromacs.org/5784

#5 - 04/07/2016 08:28 AM - Gerrit Groenhof

Hi,

For ORCA related issues, please contact Christoph Riplinger, who wrote the interface and is also a ORCA developer.

Gerrit

#6 - 07/09/2016 07:11 PM - Erik Lindahl

Unfortunately seems to be very little interest from the QM/MM developers in actually looking into this.

Gerrit: While I understand you might not have written the interface, we simply don't have the resources to start maintaining the files for the QM/MM developers and coordinate contacts with the people who might have originally written it.

Unless somebody addresses it, the simple solution for us is unfortunately that we'll have to deprecate it and remove some QM/MM stuff from future releases.

#7 - 07/10/2016 09:26 AM - Gerrit Groenhof

Agreed Erik. Because we're not using the ORCA interface ourselves, there is indeed little priority to maintain from our side. Since all QM/MM can be done with a gaussian script, I can maintain that until we have a general API for doing QM/MM.

#8 - 10/25/2018 09:34 AM - Mark Abraham

- Related to Task #2706: Rework classic QM/MM interface added

Files

pyp.mdp	1.32 KB	03/31/2016	Grzegorz Wieczorek
example.ORCAINFO	249 Bytes	03/31/2016	Grzegorz Wieczorek
qm_orca.c	13.5 KB	04/01/2016	Grzegorz Wieczorek
qm_orca.c	13.5 KB	04/01/2016	Grzegorz Wieczorek
qm_orca.c	13.9 KB	04/01/2016	Grzegorz Wieczorek