

GROMACS - Bug #287

SIGILL crash in nb_kernel with user tables and energygrp_table

02/04/2009 09:35 PM - Janne Blomqvist

Status: Closed	
Priority: Normal	
Assignee: Erik Lindahl	
Category: mdrun	
Target version: 4.0	
Affected version - extra info:	Difficulty: uncategorized
Affected version:	
Description	
Hi,	
I'm using Gromacs 4.0.2 on CentOS 5 x86-32 using the packages from the EPEL repository. Gromacs usually works, but when I set up a system with user tables and energygrp_table option for nonbonded interactions it crashes. Running (g_)mdrun via gdb shows:	
Program received signal SIGILL, Illegal instruction. [Switching to Thread -1208162624 (LWP 19631)] 0x001f21a0 in nb_kernel_ia32_3dnow_test_asm () from /usr/lib/libgmx.so.5 (gdb) bt	
#0 0x001f21a0 in nb_kernel_ia32_3dnow_test_asm () from /usr/lib/libgmx.so.5	
#1 0x001f759d in nb_kernel_ia32_3dnow_test () from /usr/lib/libgmx.so.5	
#2 0x001f75ce in nb_kernel_setup_ia32_3dnow () from /usr/lib/libgmx.so.5	
#3 0x0019243c in gmx_setup_kernels () from /usr/lib/libgmx.so.5	
#4 0x00d40881 in init_forcerec () from /usr/lib/libmd.so.5	
#5 0x08055cac in do_cg ()	
#6 0x0805b59a in update_annealing_target_temp@plt ()	
#7 0x00597dec in __libc_start_main () from /lib/libc.so.6	
#8 0x0804b021 in do_cg ()	
Looking at /proc/cpuinfo, we see that my machine doesn't have 3dnow which presumably is the reason for the crash:	
processor : 0 vendor_id : GenuineIntel cpu family : 6 model : 13 model name : Intel(R) Pentium(R) M processor 1.60GHz stepping : 8 cpu MHz : 600.000 cache size : 2048 KB fdiv_bug : no hlt_bug : no f00f_bug : no coma_bug : no fpu : yes fpu_exception : yes cpuid level : 2 wp : yes flags : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat clflush dts acpi mmx fxsr sse sse2 ss tm pbe nx up est tm2 bogomips : 1197.82	

History

#1 - 02/04/2009 11:12 PM - Erik Lindahl

Hi,

This isn't a/the bug you're seeing, but Gromacs detection of CPU capabilities. We set a posix longjmp instruction to capture SIGILL, and then try to execute a 3DNow (and subsequently SSE) instruction to see if it works. If not, the longjmp is executed and we proceed.

However, when you run this in GDB the debugger catches the signal first, before we can intercept it.

If you cannot get it to continue you might unfortunately have to build a gmx version from source where you disable 3dnow support to get past this in the debugger.

#2 - 02/05/2009 12:05 AM - Jussi Lehtola

The Fedora RPMs don't contain debug-friendly binaries for non-bonded instructions, so you will have to compile gromacs with the needed additional flags yourself.

Also, just shooting in the dark: you might want to try updating to 4.0.3 with the EPEL testing repository enabled (# yum --enablerepo=epel-testing update gromacs*) and see if it fixes the bug.

#3 - 02/05/2009 07:58 PM - Janne Blomqvist

(In reply to comment #1)

Hi,

This isn't a/the bug you're seeing, but Gromacs detection of CPU capabilities. We set a posix longjmp instruction to capture SIGILL, and then try to execute a 3DNow (and subsequently SSE) instruction to see if it works. If not, the longjmp is executed and we proceed.

Ah, thanks for the explanation, and sorry for what appears to be a non-bug. As an aside, why not use the "cpuid" instruction to get the features that the cpu supports directly rather than trying to execute potentially invalid instructions? See

<http://en.wikipedia.org/wiki/CPUID>

Anyways, I tried my model on an Opteron machine where 3dnow is available, and Gromacs 4.0.3 crashed with

```
Program received signal SIGSEGV, Segmentation fault.
[Switching to Thread 182968939264 (LWP 16561)]
0x00000000005ff09a in _nb_kernel030_x86_64_sse.nb030_dosingle ()
(gdb) bt
#0 0x00000000005ff09a in _nb_kernel030_x86_64_sse.nb030_dosingle ()
```

I suspect that my input files might be messed up, so I'll see if I can find the problem myself before whining more about this issue. I'm actually trying to port a coarse-grained polymer model from another simulation program to Gromacs, so this is all a bit new for me.

#4 - 02/05/2009 08:12 PM - Erik Lindahl

Hi,

There are deep historical reasons behind the CPU detection. The problem is that CPUID only reports on the *hardware* support for e.g. SSE; not whether the operating system correctly saves the registers in question on a context switch.

This was a huge problem when SSE first appeared - everything seemed to work fine until you got a context switch in the middle of an SSE loop, and then you were screwed. There were actually other registers (or instructions, I forget...) you can use to check if the software context-switch support for SSE was present, but those could only be executed with root permission.

Thus we ended up with the solution where we had to "test execute" an instruction to see if it worked!

However, this is no longer a serious issue since any OS post-2002 should support it just fine, so I've already replaced it with `_cpuid` calls in my local code. There's a bunch of other changes (and remaining bugs) there, so it will be a week or so before that shows up in cvs head.

My gut feeling for your error would be that your coordinates are screwed up so you try to access table data beyond the end of the allocated memory. It might be easier to debug in the C kernels, which you force by setting the environment variable `NOASSEMBLYLOOPS` to any value.

#5 - 02/06/2009 11:33 AM - Berk Hess

We should really add a `-checkdist` option to `mdrun` that checks if interactions are beyond the table length or if they are a certain distance beyond the cut-off. I have a partial version in my local code and it has already proven very useful for tracking down the source of crashes.

Berk

#6 - 02/12/2009 11:23 AM - Janne Blomqvist

Hi,

just a quick update. It seems my problem was indeed with the input data, and not with gromacs itself. I suspect the reason for the crash was that during the initial equilibration with steepest descent the nonbonded force on some bead overflowed to infinity (which happens pretty easily with single precision when you're used to double!). Using force capping for both the force and potential (according to the scheme in Auhl et al., JCP 119, 12718) solved this. I had originally planned to use force capping only once I started with the md runs, but it turned out to be necessary for steepest descent as well.

Thanks again for the quick help.

#7 - 02/12/2009 11:49 AM - Berk Hess

This was not a bug.

But it would be very helpful to have some distance checking option in mdrun.

Berk