

## GROMACS - Bug #341

### append option continuing to write to previous trajectory crashed when dealing with a large traj.trr file

07/20/2009 10:17 PM - ckcumaa empty

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> Erik Lindahl	
<b>Category:</b> mdrun	
<b>Target version:</b> 4.0_rc1	
<b>Affected version - extra info:</b>	<b>Difficulty:</b> uncategorized
<b>Affected version:</b>	

#### Description

Overview:

"append" command to continue to write to the previous trajectory file crashes once the traj.trr file becomes larger than 2GB in Gromacs 4.0.5.

I've previously reported this bug when I tested with version 4.0.4 (Please check the bug report [#315](#))

The same problem still exists and I cannot append the simulation once the trajectory file becomes larger than 2GB. I tested on different distributions of linux, such as redhat and suse, but the same problem had occurred.

I guess this is a major bug of all new version of gromacs (after 4.0.2)

You asked me before if the problem is related with the file server that does not support large files but it definitely has nothing to do with that issue. I also tested on my local machine but I could see the same problem.

However, the version 4.0.2 does not have any problem in continuing a simulation with "append" option. I guess one of the updates from v4.0.2 to v4.0.3 cause the problem.

I've been spending a lot of time on debugging the problem but I didn't find the answer yet.

I hope you will check this issue in detail. I really need to use new version of gromacs later than 4.0.4. for my research.

If you try to append the simulation when the trajectory file is slightly larger than 2GB, it may be working. Please try with a larger trajectory file such as over 5GB then you will see the problem.

I've checked that the bug report [#315](#) still have the input files to reproduce the bug so I think you can test the bug with those files.

#### History

##### #1 - 08/31/2009 12:10 PM - Berk Hess

I didn't realize this bug report was here, while we fixed something at least related.

Any 4.0 version could not append to any file larger than 2GB.

The fix is one line in src/gmxlib/checkpoint.c:

```
-      outputfiles[i].offset = ( ((off_t) offset_high) << 32 ) | ( (off_t) offset_low );  
+      outputfiles[i].offset = ( ((off_t) offset_high) << 32 ) | ( (off_t) offset_low & mask );
```

Could you check if this fixes your problem and also bug 315?

Berk

##### #2 - 08/31/2009 07:01 PM - ckcumaa empty

(In reply to comment [#1](#))

I didn't realize this bug report was here, while we fixed something at least related.

Any 4.0 version could not append to any file larger than 2GB.

The fix is one line in src/gmxlib/checkpoint.c:

```
-         outputfiles[i].offset = ( ((off_t) offset_high) << 32 ) | ( (off_t) offset_low );  
+         outputfiles[i].offset = ( ((off_t) offset_high) << 32 ) | ( (off_t) offset_low & mask );
```

Could you check if this fixes your problem and also bug 315?

Berk

Thank you for your advice.

I've tested version 4.0.5 with your recommendation but it didn't work for me.

Since version 4.0.2 is working well even without the modification of the checkpoint.c, I think it's a different problem.

Please let me know if you have any other advice.

### **#3 - 09/01/2009 09:59 AM - Berk Hess**

I can not reproduce your problem.

For me appending to a 5 GB trr file works fine in 4.0.5 with the fix of my previous reply.

Could you run `gmxdump -cp` and report the file offset for the trr?

For about 5GB I get:

output filename = traj.trr

file\_offset\_high = 1

file\_offset\_low = 1108032704

What operating system are you running on?

Note that this bug is not critical, since you can run without `-append` and use `trjcat`.

Berk

### **#4 - 09/01/2009 10:13 AM - Berk Hess**

I can add that the error message of truncate you report in bugzilla 315: "invalid arguments" is exactly what would happen with any Gromacs 4.0.6 version when your file size would be between 2 and 4 GB or between 6 and 8 GB, etc. This corresponds to a negative "file\_offset\_low" as reported by `gmxdump cp`.

The fix I mailed fixes this issue.

Nothing essential seems to have changed between 4.0.2 and 4.0.3.

Could it be that you coincidentally checked 4.0.2 with a trr size of between 4 and 6 GB, where you checked 4.0.3 with a size between 2 and 4 GB or between 6 and 8 GB?

Are you really sure you recompiled `gmxlib` and `mdrun` after applying the code change I mailed you?

I would guess that your problem is caused by the bug we fixed.

Berk

### **#5 - 09/01/2009 10:14 AM - Berk Hess**

In my previous reply I meant to say any Gromacs 4.0.? version. Gromacs 4.0.6 will include the fix.

Berk

### **#6 - 09/01/2009 08:15 PM - ckcumaa empty**

(In reply to comment [#4](#))

I've tested "append" option with version 4.0.2 and 4.0.5 without the modification of the checkpoint.c file. I restarted my job when the traj.trr is 3GB, 7GB, and 12GB.

For version 4.0.2, there was no problem in restarting at all even though I had not modified the checkpoint.c file.

However, for version 4.0.5, the restarts were all failed for three cases and it was same even after modifying the checkpoint.c file.

The simulation could not restart even when the traj.trr is not in the range of 2-4GB and 6-8GB once I used version 4.0.x (except for 4.0.2).

Could you let me know any other, even if it has really minor effect, changes in using checkpoint file to append simulation between version 4.0.2 and others?

**#7 - 09/01/2009 09:02 PM - Berk Hess**

Could you answer the other question I had:

Could you run `gmxdump -cp` on the `cpt` file of the simulation that you can not append and report the file offset for the `trr`?

For about 5GB I get:

```
output filename = traj.trr
file_offset_high = 1
file_offset_low = 1108032704
```

Berk

**#8 - 09/01/2009 10:06 PM - ckcumaa empty**

Here's the result (version 4.0.5 and modified the `checkpoint.c` file)

1. when `traj.trr` is 1.5 GB (append option was working)

```
output filename = traj.trr
file_offset_high = 0
file_offset_low = 1361556000
```

2. when `traj.trr` is 3.0 GB (append option was not working)

```
output filename = traj.trr
file_offset_high = 0
file_offset_low = -1717318592
```

3. when `traj.trr` is 7.0 GB (append option was not working)

```
output filename = traj.trr
file_offset_high = 0
file_offset_low = -1171197888
```

4. when `traj.trr` is 11.0 GB (append option was not working)

```
output filename = traj.trr
file_offset_high = 0
file_offset_low = -2014065888
```

**#9 - 09/01/2009 10:12 PM - Berk Hess**

Ah, that very interesting.

Are these the results of `gmxdump` on the `trr` before appending?  
I guess the `trr` file does not change if the truncate fails.

So `offset_high` is always 0, even though it should be 1 for files larger than 4 GB.

Can you report the `gmxdump` offset output for a 4.0.2 checkpoint file that did work for a size between 4 and 6 GB?

Thanks,

Berk

**#10 - 09/01/2009 10:48 PM - ckcumaa empty**

(In reply to comment [#9](#))

Here's another result (version 4.0.2)

1. when `traj.trr` is 1.5 GB (append option was working)

```
output filename = traj.trr
file_offset_high = 0
file_offset_low = 1383168000
```

2. when `traj.trr` is 3.0 GB (append option was working)

```
output filename = traj.trr
file_offset_high = 0
file_offset_low = -1442183296
```

3. when `traj.trr` is 5.0 GB (append option was working)

```
output filename = traj.trr
file_offset_high = 0
file_offset_low = 922169504
```

4. when `traj.trr` is 7.0 GB (append option was working)

output filename = traj.trr  
file\_offset\_high = 0  
file\_offset\_low = -1432703088

5. when traj.trr is 11.0 GB (append option was working)

output filename = traj.trr  
file\_offset\_high = 0  
file\_offset\_low = -1947731584

Thank you

**#11 - 09/02/2009 09:52 AM - Berk Hess**

I see what the problem is now.  
The off\_t value does not get converted correctly into two ints.  
The high part is always 0.

Appending "worked" in 4.0.2, because we only check the return value of truncate since 4.0.3. This means that probably in 4.0.2 your trr file did not get truncated at all and you might have double frames.

The question is now why it goes wrong.  
One option could be that we check the size of off\_t against the size of int, whereas it should have been checked against 4.

I asked you before what system you are running on, could you tell me this?

Could you compile the small C program below with the same compiler options as Gromacs and run it and report the output value?

If this gives 8, then changing lines 780 and 792 of src/gmxlib/checkpoint.c

```
from
#if (SIZEOF_OFF_T > SIZEOF_INT)
to
#if (SIZEOF_OFF_T > 4)
should fix the problem, if I did not overlook anything.
```

Thanks,

Berk

```
#include <stdio.h>
```

```
int main() {
printf("%d\n", sizeof(int));
```

```
return 0;
}
```

**#12 - 09/02/2009 05:34 PM - ckcumaa empty**

(In reply to comment [#11](#))

First of all, I really appreciate your help.

I had compiled your C code and it returned 4 instead of 8.  
and the Red Hat Enterprise Linux 5 is the operating system.

Thank you  
kyungchan

**#13 - 09/02/2009 06:18 PM - Berk Hess**

OK, so the int size does not seem to be the issue.  
To check what actually happens in the Gromacs compilation,  
could you do:  
grep SIZEOF src/config.h  
in the directory where you compiled Gromacs and post the results?

```
On my workstation I get:
#define SIZEOF_INT 4
#define SIZEOF_LONG_INT 8
#define SIZEOF_LONG_LONG_INT 8
#define SIZEOF_OFF_T 8
```

Berk

**#14 - 09/02/2009 07:06 PM - ckcumaa empty**

(In reply to comment [#13](#))

I got:

```
#define SIZEOF_INT 4
#define SIZEOF_LONG_INT 4
#define SIZEOF_LONG_LONG_INT 8
#define SIZEOF_OFF_T 4
```

Thank you

**#15 - 09/02/2009 07:24 PM - Berk Hess**

Ah, that's it!

The size of off\_t is reported as 4.

This means you can not truncate files larger than 4 GB (or maybe 2 GB).

Before you mailed that the system administrator checked that the the size of off\_t is a bytes. But the configure script of Gromacs detects it differently.

I don't know if the size of off\_t depends on some settings. My workstation gives 8, my netbook gives 8 in the Gromacs config.h, but gives 4 when I run the program below.

I'll discuss this with Erik tomorrow.

We should at least add a flag in the .cpt file to not append and give a warning when sizeof(off\_t)=4.

Berk

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
```

```
int main() {
printf("%d\n",sizeof(off_t));
```

```
return 0;
}
```

**#16 - 09/02/2009 07:29 PM - Berk Hess**

Having off\_t only 4 bytes sounds strange though. What kind of architecture/processor do you have?

Berk

**#17 - 09/02/2009 08:38 PM - ckcumaa empty**

(In reply to comment [#16](#))

For cluster,

I'm using dual ~ quad core AMD opteron 3.0 GHz cpus and system memory per node ranges from 2GB to 32GB. The redhat linux enterprise 5.0 is operating system.

F.Y.I

The following is the answer from our system administrator [It is a 64 bit system, but it is a LP64 system by default, that is LONG and POINTER is 8 bytes (64 bit) int is still 4 bytes (32 bit)]

For my local machine, It's a 32 bit system and the operating system is the red hat linux.

kyungchan

**#18 - 09/03/2009 06:15 PM - Berk Hess**

I just put in a proper check for sizeof(off\_t)=4 and appending of files > 2 GB for the Gromacs 4.0.6 release. This does not mean it appending of large files will work on your system then, but you will get a proper error message.

Why your system does not have 8 byte off\_t size is another matter. You hardware is quite new and your operating system also seems quite new.

Maybe configure/autoconf does not detect properly that large file support can be turned on.

Berk

**#19 - 09/04/2009 03:29 AM - ckcumaa empty**

(In reply to comment [#18](#))

Thank you Berk for all your help.  
I'll test version 4.0.6 and report the result.

kyungchan

**#20 - 09/04/2009 10:06 AM - Berk Hess**

I looked at your replies once more and now noticed that you are talking about two different machines: your workstation and a cluster.

Now I don't understand on which machine you are having the appending problem, on which machine you compiled the binary that gives the problem and on which machines you did the checks.

You post with SIZEOF\_LONG=4 shows that at least on that machine your compiler is configured for 32 bit pointers. Note that this does not imply anything about the size of off\_t (which was also 32 bit in your post).

Berk

**#21 - 09/04/2009 07:10 PM - ckcumaa empty**

(In reply to comment [#20](#))

OK, the previous report was from my local machine and the following results are from our system administrator and he got on the cluster.

```
#define SIZEOF_INT 4
#define SIZEOF_LONG_INT 8
#define SIZEOF_LONG_LONG_INT 8
#define SIZEOF_OFF_T 8
```

The OFF\_T values are exactly same sa yours so I'll test on our cluster with the modification of checkpoint.c and report the results.

Thank you for your advice.

kyungchan

**#22 - 09/04/2009 07:18 PM - Erik Lindahl**

Hi,

I think we've pretty much narrowed it down to the off\_t, but we're still a bit perplexed why it's not identified correctly on your cluster. As Berk said, we will "fix" this by making sure to disable a lot of the appending when we cannot find 64-bit support, but it would of course be good to fix the detection so it works on all systems too for the future.

1) Do you know how gromacs was compiled on your system?

2) If not, could you possibly help us to test two things:

2a) Send us the output of "autoconf --version" and "automake --version"

2b) Download the gromacs source (any 4.x.y version is fine), run the ./configure script, and send us the two files "config.log" and "src/config.h"?

That might help us to debug the detection - unless it turns out to be a problem on your cluster, of course ;-)

Cheers,

Erik

**#23 - 09/04/2009 08:02 PM - ckcumaa empty**

(In reply to comment [#22](#))

I got

GNU Autoconf 2.5.9  
GNU Automake 1.9.6

I've sent you config.log and config.h to your email account

Thank you.

kyungchan

**#24 - 09/09/2009 03:49 AM - ckcumaa empty**

(In reply to comment [#22](#))

I've tested the version 4.0.5 on the clusters of my university, which has the same OFF\_T values as yours, with the modification of checkpoint.c file and it's working fine now.

There's no problem in using the "append" option on 64bit linux machine now.

I really appreciate all of your advice.

kyungchan