GROMACS - Bug #1904
pair search over-splitting
02/15/2016 10:03 PM - Szilárd Páll

Status: Closed
Priority: Normal
Assignee: Szilárd Páll
Category: mdrun
Target version: 5.1.3
Affected version - extra info:
Affected version: 5.1.2

Difficulty: uncategorized

Description
With a large target super-cluster count and large input size the list splitting code can overflow resulting in excessive list splitting. This causes considerable performance degradation in the non-bonded kernel. It is more prone to happen with GPUs with large processor count e.g. AMD Fiji with 64 proc where min_ci=3200.

Repro, 192k water box, running on AMD R9 Fury Nano, nstlist=25

<table>
<thead>
<tr>
<th>min_ci</th>
<th>kernel (ms)</th>
<th>#super clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3.498</td>
<td>4519</td>
</tr>
<tr>
<td>3000</td>
<td>5.437</td>
<td>95906</td>
</tr>
<tr>
<td>3200 (auto)</td>
<td>5.436</td>
<td>95906</td>
</tr>
</tbody>
</table>

Associated revisions
Revision f7c18fba - 02/19/2016 09:32 AM - Berk Hess
Fix excessive list splitting

Due to a possible integer overflow, the pair list splitting code could end up over-splitting pair lists and causing large performance degradation. Due to the larger processor count, runs using AMD GPUs, using 100k+ simulation systems are more prone to suffer from the issue.

Fixes #1904
Change-Id: I29139ec80aa75c78fa93de085f7c60cdae88d5b

History
#1 - 02/15/2016 10:06 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1904.
Uploader: Szilárd Páll (pall.szilard@gmail.com)
Change-Id: I29139ec80aa75c78fa93de085f7c60cdae88d5b
Gerrit URL: https://gerrit.gromacs.org/5642

#2 - 02/15/2016 10:26 PM - Berk Hess
Although I see that is a risk of overflow here, I don't understand how a large min_ci increases the chance of this happening. I would rather think that a large min_ci lowers the risk.

#3 - 02/15/2016 10:59 PM - Szilárd Páll
Berk Hess wrote:

Although I see that is a risk of overflow here, I don't understand how a large min_ci increases the chance of this happening. I would rather think that a large min_ci lowers the risk.

Running GMX_NB_MIN_CI=3200 $gmx mdrun -nstlist 25 with the 192k water input gives: nsp_tot_est=1986914, nsp_target_av=621; the product of these two values is already close to INT_MAX, and with the factor of 3 it overshoots nicely.

You're right, min_ci comes in only through the j-list length estimate and even there in a denominator so that would mean that this overflow could happen with smaller min_ci too, perhaps even on a TITAN with 40*24=960?

02/18/2020
nsp_tot_est is independent of min_ci and nsp_target_av is inversely proportional to min_ci, so I would think the GPU with the fewest SMXs would be most prone to this issue, not the Titan.

PS I uploaded a proper fix that keeps nsp_tot_est as a float and thus the overflow can no longer occur.

I overlooked the check at the start of get_nsubpair_target that skips the balancing when the number of grid cells is larger than the number of list we want. (I recalled there was such a check, but I'm not so sharp with my fever). That explains why you need a large min_ci (and a large, but not too large system).

I also overlooked that. This code could use a bit of refactoring - or documentation, not sure -, it's quite hard to get what's going on here. I'll make an attempt to do something in a new change, but your input will probably be needed.

I don't think there's much refactoring to be done here. I added some minor clarification to the documentation in "my" change. I also realized we should get rid of the term "subpair" here and change it to cluster, but I'd rather to that extensive renaming in master.

By "here" I meant in the code-path I have been looking at while debuggin the issue, all the way from make_nbnxn_pairlist() which calls get_nsubpair_target() to the splitting code.

BTW, interestingly, even though the search gives 40% more lists than min_ci even without splitting, the estimate is off by enough to trigger splitting and end up with 65% more ci's than requested (5281/3200).

Note: we concluded that this has most likely affected only AMD GPUs, but on the larger Hawaii or Fiji devices it was not unlikely to cause a large performance drop.