Gromacs - Feature #1188

Separate timing reports for PP and PME nodes

03/12/2013 01:59 PM - Mikhail Plotnikov

Status: Closed
Priority: Normal
Assignee: Mark Abraham
Category: mdrun
Target version: 5.0

Description
Please implement separate timing reports for PP and PME nodes in case separate PME nodes are used (-npme option). Current report is a mess in this case, because it contains PP and PME timings and thus total walltime is smaller than sum of all components. Also PME thread count is incorrectly reported and probably timing itself is incorrect in case PME num threads is taken from OMP_NUM_THREADS variable and is not set by –ntomp_pme option.
To reproduce please use attached test case.

Associated revisions
Revision 1df1ceb2 - 02/03/2014 02:02 AM - Mark Abraham
Improved wallcycle reporting

Removed reporting of MPI and thread counts on each row, in favour of a header with that information.

With npme > 0, prints a note that the time column is not supposed to add up.

Works correctly with a range of -npme, -ntomp and -ntomp_pme values: PP times add up to the total, which equals the final walltime reported; cycle count and percentage column totals are correct and reflect the actual work done.

Partial fix for #1188

Change-Id: lc870d981bf0375189601bf8c9bc67bc5d626497

History
#1 - 04/30/2013 09:19 PM - Mark Abraham
- Status changed from New to Accepted
- Assignee set to Mark Abraham
- Target version changed from 4.6.2 to 4.6.3
- Affected version set to 4.6

Reproduced, thanks.

In the pre-threading era we produced output like:

11/25/2015
<table>
<thead>
<tr>
<th>Computing:</th>
<th>Nodes</th>
<th>Number</th>
<th>G-Cycles</th>
<th>Seconds</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain decomp.</td>
<td>48</td>
<td>750</td>
<td>154.369</td>
<td>220.5</td>
<td>1.9</td>
</tr>
<tr>
<td>DD comm. load</td>
<td>48</td>
<td>750</td>
<td>1.186</td>
<td>1.7</td>
<td>0.0</td>
</tr>
<tr>
<td>DD comm. bounds</td>
<td>48</td>
<td>751</td>
<td>10.257</td>
<td>14.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Send X to PME</td>
<td>48</td>
<td>7501</td>
<td>36.525</td>
<td>52.2</td>
<td>0.4</td>
</tr>
<tr>
<td>Comm. coord.</td>
<td>48</td>
<td>7501</td>
<td>167.265</td>
<td>239.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Neighbor search</td>
<td>48</td>
<td>751</td>
<td>345.441</td>
<td>493.5</td>
<td>4.2</td>
</tr>
<tr>
<td>Force</td>
<td>48</td>
<td>7501</td>
<td>4363.061</td>
<td>6233.0</td>
<td>53.4</td>
</tr>
<tr>
<td>Wait + Comm. F</td>
<td>48</td>
<td>7501</td>
<td>370.087</td>
<td>528.7</td>
<td>4.5</td>
</tr>
<tr>
<td>PME mesh</td>
<td>16</td>
<td>7501</td>
<td>1652.623</td>
<td>2360.9</td>
<td>20.2</td>
</tr>
<tr>
<td>Wait + Comm. X/F</td>
<td>16</td>
<td></td>
<td>388.828</td>
<td>555.5</td>
<td>4.8</td>
</tr>
<tr>
<td>Wait + Recv. PME F</td>
<td>48</td>
<td>7501</td>
<td>273.452</td>
<td>390.6</td>
<td>3.3</td>
</tr>
<tr>
<td>Write traj.</td>
<td>48</td>
<td>16</td>
<td>2.632</td>
<td>3.8</td>
<td>0.0</td>
</tr>
<tr>
<td>Update</td>
<td>48</td>
<td>7501</td>
<td>87.795</td>
<td>125.4</td>
<td>1.1</td>
</tr>
<tr>
<td>Constraints</td>
<td>48</td>
<td>7501</td>
<td>112.100</td>
<td>160.1</td>
<td>1.4</td>
</tr>
<tr>
<td>Comm. energies</td>
<td>48</td>
<td>752</td>
<td>126.219</td>
<td>180.3</td>
<td>1.5</td>
</tr>
<tr>
<td>Rest</td>
<td>48</td>
<td>73.972</td>
<td>105.7</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

| Total        | 64    |     | 8165.812 | 11665.5 | 100.0 |

i.e. the columns added correctly, but we did not separate PP-only from PME-only when that was what was going on. I am working on a patch to return to that functionality.

#2 - 04/30/2013 09:20 PM - Mark Abraham
- Category set to mdrun
- Status changed from Accepted to In Progress

#3 - 04/30/2013 10:03 PM - Mark Abraham
- Status changed from In Progress to Fix uploaded

I have not been able to see any issues with OMP_NUM_THREADS such as Mikhail reported. Reported thread counts and timings look correct to me.

https://gerrit.gromacs.org/2352

#4 - 05/10/2013 03:28 PM - Berk Hess
- Status changed from Fix uploaded to In Progress
The timing reports are separate, but written out to the same table, which can be confusing.
I have a patch in progress which writes two separate tables when separate PME ranks are used.
I guess with OMP_NUM_THREADS Mikhail actually means have different thread counts on different ranks.

#5 - 05/31/2013 03:22 PM - Mark Abraham
Moved discussion here from https://gerrit.gromacs.org/2412

There, Berk and I were discussing the merits of starting iteration timers in sync or not, and how to interpret the numbers that result.

I don't feel synchronization is a goal we should have. MPMDism and workload overlap is going to get more complex. Already we have the first iteration of PP starting later than the first iteration of PME, because the former has more setup after the latter gets its first data. That only affects the timing of the first iteration, which is no big deal.

I think one of the things that has to break as we move towards more task parallelism is the notion of homogeneity in the role of a set of processing units. I think the main iteration loop should look the same on each process, and the data it has should determine what work it should do. That work can get billed to different accounts so we know where things stand, sure.

In this light, the proposed master-branch split of the wallcycle-reporting table into PP and PME sections seems like a backward step. Why shouldn't the table look the same for npme=0 and npme>0? Right now, in choosing which npme value to use, the important thing is the total time. Tweaking the settings will lead to changes in half a dozen of the table rows. Those changes have useful feedback, sure. I do not see what is sacred about the walltime that we should scale raw counts and not the times reported. They're both scaled from the same value! Knowing the PME mesh time relative to the time spent by some number of PME-only nodes is no more useful than knowing the PME mesh time relative to the time spent by all nodes. And it's less useful for evaluating the particular case people have at the moment of whether to use the MPMD, because the timings change scale when you do so.

#6 - 06/26/2013 12:32 AM - Mark Abraham
- Target version changed from 4.6.3 to 4.6.x

#7 - 07/08/2013 11:18 AM - Szilárd Páll

Mark Abraham wrote:

I don't feel synchronization is a goal we should have. MPMDism and workload overlap is going to get more complex. Already we have the first iteration of PP starting later than the first iteration of PME, because the former has more setup after the latter gets its first data. That only affects the timing of the first iteration, which is no big deal.

I agree, but I think we should not change the synchronous timing behaviour of GROMACS in a patch release. Hence, any such changes should go into 5.0.

I think one of the things that has to break as we move towards more task parallelism is the notion of homogeneity in the role of a set of processing units. I think the main iteration loop should look the same on each process, and the data it has should determine what work it should do. That work can get billed to different accounts so we know where things stand, sure.

In this light, the proposed master-branch split of the wallcycle-reporting table into PP and PME sections seems like a backward step. Why shouldn't the table look the same for npme=0 and npme>0? Right now, in choosing which npme value to use, the important thing is the total time. Tweaking the settings will lead to changes in half a dozen of the table rows. Those changes have useful feedback, sure. I do not see what is sacred about the walltime that we should scale raw counts and not the times reported. They're both scaled from the same value! Knowing the PME mesh time relative to the time spent by some number of PME-only nodes is no more useful than knowing the PME mesh time relative to the time spent by all nodes. And it's less useful for evaluating the particular case people have at the moment of whether to use the MPMD, because the timings change scale when you do so.
the timings change scale when you do so.

My impression is that a much more bottom-up redesign of the cycle-timing is required with a full task parallelization. In particular, not only MPMD needs to be considered, but tasks (dispatchers?) will need to record their own timing and the thread/task manager will have to do the accounting per task and per rank.

The incorrect thread count was already fixed in 4.6.2. The table has not been split, since that makes it more difficult to see what percentage of the resources was spent on what part of the calculations. But in 5.0 the separate PME node entries are marked with an asterisk and a note on the total time has been added.

Files
PME_timings_report.tbz2 6.84 MB 03/12/2013 Mikhail Plotnikov