Occupancy studies

02-14-2022
Module occupancies

- Change pixel module occupancy allocations. However, the way it stored was as an “extra module” append to md/sg module occupancies, still no significant improvements.

<table>
<thead>
<tr>
<th>Per module</th>
<th>Max pixel md</th>
<th>Max pixel sg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous allocated</td>
<td>100000</td>
<td>50000</td>
</tr>
<tr>
<td>Actually used</td>
<td>40080</td>
<td>20040</td>
</tr>
<tr>
<td>Updated allocated</td>
<td>45000</td>
<td>25000</td>
</tr>
</tbody>
</table>
99.99% quantile test

- only changing t3 and t5 occupancy
Module occupancies

Put a memory cut off at 99.99% quantile of modules occupancy. That means, we don’t store modules that have more than that occupancy.
Max Memory use after changing

This plot shows the memory use when monitoring running four different versions of PU200 events after changing the pT3, pT5 module occupancy allocation (in both allocation and writing plots). That’s much improvement to conclude. The yellow number changes because this version is with 62 PRs merging in. (Based on a newer version on Github)

* units in MB, x-axis is the percentage of memory use. The orange arrow is the original memory usage. The blue one is choosing 99.99% quantile as the allocated memory and the new memory usage. (Total memory: 32510)
Before and after the T3 and t5 occupancy change, explicit PU200

No significant efficiency drop is seen after the 99.99% truncation of the t3 and t5 occupancy
This is the timing result. (With Max_Trituplet_module and Max_Quintuplet_module changed to 99.99% quantile). PU200 explicit, 200 events in total.

<table>
<thead>
<tr>
<th>Event (Evt)</th>
<th>Hits</th>
<th>MD</th>
<th>LS</th>
<th>T3</th>
<th>T5</th>
<th>pLS</th>
<th>pT5</th>
<th>pT3</th>
<th>TC</th>
<th>TCE</th>
<th>Event</th>
<th>Short</th>
<th>Loop</th>
<th>Effective</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>301.</td>
<td>4.</td>
<td>14</td>
<td>15</td>
<td>22</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>7</td>
<td>18</td>
<td>409.</td>
<td>99.+/-25</td>
<td>418.</td>
<td>102. explicit[s=1]</td>
</tr>
<tr>
<td>avg</td>
<td>311.</td>
<td>7.</td>
<td>16</td>
<td>19</td>
<td>28</td>
<td>9</td>
<td>14</td>
<td>12</td>
<td>9</td>
<td>20</td>
<td>444.</td>
<td>124.+/-44</td>
<td>230.</td>
<td>64. explicit[s=2]</td>
</tr>
<tr>
<td>avg</td>
<td>339.</td>
<td>13.</td>
<td>23</td>
<td>29</td>
<td>40</td>
<td>10</td>
<td>27</td>
<td>24</td>
<td>18</td>
<td>26</td>
<td>550.</td>
<td>201.+/-65</td>
<td>145.</td>
<td>53. explicit[s=4]</td>
</tr>
<tr>
<td>avg</td>
<td>373.</td>
<td>23.</td>
<td>32</td>
<td>43</td>
<td>60</td>
<td>13</td>
<td>53</td>
<td>46</td>
<td>52</td>
<td>43</td>
<td>738.</td>
<td>352.+/-108</td>
<td>133.</td>
<td>63. explicit[s=6]</td>
</tr>
<tr>
<td>avg</td>
<td>402.</td>
<td>30.</td>
<td>41</td>
<td>59</td>
<td>86</td>
<td>19</td>
<td>87</td>
<td>66</td>
<td>97</td>
<td>66</td>
<td>952.</td>
<td>531.+/-133</td>
<td>126.</td>
<td>70. explicit[s=8]</td>
</tr>
</tbody>
</table>
99% quantile test

-only changing t3 and t5 occupancy
Put a memory cut off at 99.99% quantile of modules occupancy. That means, we don’t store modules that have more than that occupancy.

Module occupancies

**Md**
- Max module occupancy: 386
- Module occupancy 90% quantile: 8.82
- Module occupancy 99% quantile: 23.08
- Module occupancy 99.99% quantile: 88.47
- Current allocated: 100

**Sg**
- Max module occupancy: 3351
- Module occupancy 90% quantile: 34.77
- Module occupancy 99% quantile: 144.05
- Module occupancy 99.99% quantile: 536.08
- Current allocated: 600

**T3**
- Max module occupancy: 4554
- Module occupancy 90% quantile: 39.28
- Module occupancy 99% quantile: 173.13
- Module occupancy 99.99% quantile: 1169.18
- Current allocated: 2500

**T5**
- Max module occupancy: 3544
- Module occupancy 90% quantile: 15.72
- Module occupancy 99% quantile: 50.53
- Module occupancy 99.99% quantile: 512.13
- Current allocated: 3000
Max Memory use after changing

This plot shows the memory use when monitoring running four different versions of PU200 events after changing the pT3, pT5 module occupancy allocation (in both allocation and writing plots). That's much improvement to conclude. The yellow number changes because this version is with 62 PRs merging in. (Based on a newer version on Github)

<table>
<thead>
<tr>
<th>Version</th>
<th>Memory Use (MB)</th>
<th>Memory Use (MB)</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td>1000</td>
<td>3476</td>
<td>3.08%</td>
</tr>
<tr>
<td>unified</td>
<td>933</td>
<td>2145</td>
<td>2.87%</td>
</tr>
<tr>
<td>explicit cache</td>
<td>1218</td>
<td>4440</td>
<td>3.75%</td>
</tr>
<tr>
<td>unified cache</td>
<td>1022</td>
<td>2551</td>
<td>3.14%</td>
</tr>
</tbody>
</table>

* units in MB, x-axis is the percentage of memory use. The orange arrow is the original memory usage. The blue one is choosing 99% quantile as the allocated memory and the new memory usage. (Total memory: 32510)
TCE Pt comparison

Before and after the T3 and t5 occupancy change, explicit PU200

Slightly drop is seen after the 99% truncation of the t3 and t5 occupancy
99.99% quantile test

-changing md, sg, t3, t5 occupancy together
Module occupancies

Put a memory cut off at 99.99% quantile of modules occupancy. That means, we don’t store modules that have more than that occupancy.
TCE Pt comparison

Before and after the md, sg, T3 and t5 occupancy change, explicit PU200

Some kind of fluctuation is significantly seen after the 99.99% truncation of the module occupancy