Integrating Globus and MapReduce for Out-of-computer Analysis of Peta-scale CFD Data

Max Hutchinson
University of Chicago
April 15th, 2015
Post-processing is a series of transforms and reductions
Post-processing is a series of transforms and reductions
Post-processing is a series of transforms and reductions
Post-processing is a series of transforms and reductions

Figure

post-processing (1st party)

next talk (3rd party)

Max Hutchinson
University of Chicago
MapReduce post-processing

\[
\text{Raw data} \xrightarrow{\sum} \text{Mapped data} = \text{Result}
\]

Super

Laptop
MapReduce post-processing

\[
\text{Super} \rightarrow \text{Raw data} \rightarrow \text{Raw copy} \rightarrow \sum \text{Mapped data} = \text{Result}
\]
MapReduce post-processing

\[
\sum_{\text{Laptop}} \begin{array}{c}
\text{Raw data} \\
\text{Raw copy} \\
\text{Mapped data}
\end{array} = \text{Result}
\]
MapReduce post-processing

Super

Raw data

\[ \sum \]

Laptop

Raw copy

\[ \sum \]

Mapped data = Result

Max Hutchinson

University of Chicago
MapReduce post-processing

\[ \sum \text{Laptop} \quad \text{Mapped data} = \text{Result} \]

Super

Raw data

\[ \vdots \]

Raw copy
MapReduce post-processing

\[
\sum_{\text{Laptop}} \text{Mapped data} = \text{Result}
\]

Super

Raw data

\[
\cdot
\]

Raw copy

Max Hutchinson
University of Chicago
MapReduce post-processing

\[
\sum_{\text{Laptop}} \begin{array}{c}
\text{Raw data} \\
\text{Raw copy}
\end{array} = \begin{array}{c}
\text{Mapped data} \\
\text{Result}
\end{array}
\]
<table>
<thead>
<tr>
<th>Capabilities</th>
<th>BW (Mbit/s)</th>
<th>Disk (GiB)</th>
<th>Wall (hr)</th>
<th>Core (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Frame</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tukey (local)</td>
<td>n/a</td>
<td>256.</td>
<td>0.56</td>
<td>9.06</td>
</tr>
<tr>
<td>Tukey (remote)</td>
<td>6000</td>
<td>16.</td>
<td>0.65</td>
<td>10.45</td>
</tr>
<tr>
<td><strong>8 Frames</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tukey (local)</td>
<td>n/a</td>
<td>4096.</td>
<td>0.66</td>
<td>85.06</td>
</tr>
<tr>
<td>Tukey (remote)</td>
<td>6000</td>
<td>16.</td>
<td>5.23</td>
<td>83.61</td>
</tr>
</tbody>
</table>

\(^1\) extrapolated
## Capabilities

<table>
<thead>
<tr>
<th>1 Frame</th>
<th>BW (Mbit/s)</th>
<th>Disk (GiB)</th>
<th>Wall (hr)</th>
<th>Core (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey (local)</td>
<td>n/a</td>
<td>256.</td>
<td>0.56</td>
<td>9.06</td>
</tr>
<tr>
<td>Tukey (remote)</td>
<td>6000</td>
<td>16.</td>
<td>0.65</td>
<td>10.45</td>
</tr>
<tr>
<td>Workstation</td>
<td>100</td>
<td>2.</td>
<td>6.22</td>
<td>&lt; 12.44</td>
</tr>
<tr>
<td>Laptop(^1)</td>
<td>15</td>
<td>2.</td>
<td>38.81</td>
<td>&lt; 77.62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8 Frames</th>
<th>BW (Mbit/s)</th>
<th>Disk (GiB)</th>
<th>Wall (hr)</th>
<th>Core (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey (local)</td>
<td>n/a</td>
<td>4096.</td>
<td>0.66</td>
<td>85.06</td>
</tr>
<tr>
<td>Tukey (remote)</td>
<td>6000</td>
<td>16.</td>
<td>5.23</td>
<td>83.61</td>
</tr>
<tr>
<td>Workstation(^1)</td>
<td>100</td>
<td>2.</td>
<td>49.79</td>
<td>&lt; 99.58</td>
</tr>
<tr>
<td>Laptop(^1)</td>
<td>15</td>
<td>2.</td>
<td>310.69</td>
<td>&lt; 621.38</td>
</tr>
</tbody>
</table>

\(^1\)extrapolated
Relative performance

<table>
<thead>
<tr>
<th>1 Frame</th>
<th>BW (MBit/s)</th>
<th>Disk (GiB)</th>
<th>Wall (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy + local</td>
<td>100</td>
<td>256.</td>
<td>6.75</td>
</tr>
<tr>
<td>Out-of-computer</td>
<td>100</td>
<td>16.</td>
<td>6.22</td>
</tr>
</tbody>
</table>

Out-of-computer faster for single-use

- Overlaps communication with computation
Thank you
Questions?

▶ https://pypi.python.org/pypi/mapcombine
▶ https://pypi.python.org/pypi/glopen
▶ https://pypi.python.org/pypi/globussh
▶ https://pypi.python.org/pypi/chest
▶ https://pypi.python.org/pypi/slict
▶ Also all on GitHub