CCL Workshop
on Scalable Scientific Computing 2016

Using Docker with GPUs

Sandra Gesing
sandra.gesing@nd.edu

20 October 2016
## State of the Art

- Large speed-up of tools on accelerated approaches compared to their CPU-counter parts

<table>
<thead>
<tr>
<th>Developer</th>
<th>Speed Up</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Massachusetts General Hospital</td>
<td>300x</td>
<td><a href="http://www.opticsinfobase.org/oe-17-22-20178">http://www.opticsinfobase.org/oe-17-22-20178</a></td>
</tr>
<tr>
<td>University of Rochester</td>
<td>160x</td>
<td><a href="http://cyberaide.googlecode.com/08-cuda-biostat/vonLaszewski-">http://cyberaide.googlecode.com/08-cuda-biostat/vonLaszewski-</a></td>
</tr>
<tr>
<td>University of Amsterdam</td>
<td>150x</td>
<td><a href="http://arxiv.org/PS_cache/arxiv/">http://arxiv.org/PS_cache/arxiv/</a></td>
</tr>
<tr>
<td>Harvard University</td>
<td>130x</td>
<td><a href="http://www.springerlink.com/content/?p=c5e96a1c73340e58a313d">http://www.springerlink.com/content/?p=c5e96a1c73340e58a313d</a></td>
</tr>
<tr>
<td>University of Pennsylvania</td>
<td>130x</td>
<td><a href="http://ic.ee.upenn.edu/abstract">http://ic.ee.upenn.edu/abstract</a></td>
</tr>
<tr>
<td>Nanyang Tech, Singapore</td>
<td>130x</td>
<td><a href="http://www.opticsinfobase.org/abstract.cfm?URI=oe-17-25-23147">http://www.opticsinfobase.org/abstract.cfm?URI=oe-17-25-23147</a></td>
</tr>
<tr>
<td>University of Illinois</td>
<td>125x</td>
<td><a href="http://www.nvidia.com/object/cuda_apps_flash_new.html?state=detailsOpen">http://www.nvidia.com/object/cuda_apps_flash_new.html?state=detailsOpen</a>; aid=c24d0f-c60-c45f9-8d57-58e9460a58f</td>
</tr>
<tr>
<td>Boise State</td>
<td>100x</td>
<td><a href="http://ccon.boisestate.edu/senocak/files/BSU_CUDA_Reader_v5.pdf">http://ccon.boisestate.edu/senocak/files/BSU_CUDA_Reader_v5.pdf</a></td>
</tr>
<tr>
<td>Florida Atlantic University</td>
<td>100x</td>
<td><a href="http://portal.acm.org/citation.cfm?id=1730836.1730839&amp;coll=GUIDE&amp;dl=ACM&amp;CFID=88441459&amp;CFTOKEN=90295264">http://portal.acm.org/citation.cfm?id=1730836.1730839&amp;coll=GUIDE&amp;dl=ACM&amp;CFID=88441459&amp;CFTOKEN=90295264</a></td>
</tr>
<tr>
<td>Cambridge University</td>
<td>100x</td>
<td><a href="http://www.wbic.cam.ac.uk/~rea1/research/AIRWC.pdf">http://www.wbic.cam.ac.uk/~rea1/research/AIRWC.pdf</a></td>
</tr>
</tbody>
</table>
State of the Art

• Accelerated resources widely available (local, campus-wide, national infrastructures like XSEDE)
but...

Uptake and utilization is not following the same pace
State of the Art

- Accelerated resources widely available (local, campus-wide, national infrastructures like XSEDE)

![Allocation Usage Rate (XD SU/Hour): by Queue](chart.png)

Resource = TACC–STAMPEDE
State of the Art

• Accelerated resources widely available (local, campus-wide, national infrastructures like XSEDE) but...

Uptake and utilization is not following the same pace

 Reasons for the underutilization lay partly on the software side with proprietary and complex interfaces for development and usage
Concept

- Two groups of developers:
  - Advanced developers optimizing code for different hardware
  - Mainstream developers searching for solutions for automatic optimization
Why Docker?

- Virtual images containing only the necessary dependencies and system tools for running applications
- Containers share the kernel of the host operating system resulting in an efficient use of space and computational resources
- The layered filesystem on which Docker images are built allows for multiple images to share any common libraries, thus minimizing disk usage. It also facilitates updates to containers as added only the added images need to be installed.
• Docker eliminates the need for a hypervisor
Proof-of-concept

• SAMPO (Scalable Agent-based Mosquito POint model)
  • Models the life cycle of malaria-vectors utilizing the OpenCL API to perform computations on available accelerators (GPU, MICs).
• GPU-BLAST
  • Basic Local Alignment Search Tool developed with CUDA
• SGEMM
  • A common matrix multiplication algorithm used as a performance benchmark.
• ...
Performance tests
Limitations

• Images can become relatively large depending on the necessary binaries, libraries, and the size of the application.

• This causes problems when images are pulled from Docker Hub; the transfer time can become unreasonably long.
Minimize Transfer Time

• Minimize the images
  • Use small base images
  • Ensure any unnecessary libraries are not being installed.

• Dockerfiles-Set of instructions used to build the image from a base image.
• Save and export commands built in to the docker platform.
Future Work

• Hopefully funded project on the concept
Acknowledgements

Co-authors (paper and proposal)
• Nicholas Haydel
• Ian Taylor
• Abdul Dakkak
• Simon Garcia de Gonzalo
• Wen-mei Hwu
• Nick Sahinidis
• Nikoloas Ploskas
• Gerhard Klimeck
• Gregory Madey
• Christopher Sweet
• James Sweet
THANK YOU

sandra.gesing@nd.edu