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

Developer	Speed Up	Reference
Massachusetts General Hospital	300x	http://www.opticsinfobase.org/o 17-22-20178
University of Rochester	160x	http://cyberaide.googlecode.cor /08-cuda-biostat/vonLaszewski-
University of Amsterdam	150x	http://arxiv.org/PS_cache/arxiv/
Harvard University	130x	http://www.springerlink.com/cor /?p=c5eead9af73340e58a313d
University of Pennsylvania	130x	http://ic.ese.upenn.edu/abstract
Nanyang Tech, Singapore	130x	http://www.opticsinfobase.org/abstract.cfm?URI=oe-17-25-23147
University of Illinois	125x	http://www.nvidia.com/object /cuda_apps_flash_new.html#state=detailsOpen; aid=c24dcc0f-c60c-45f9-8d57-588e9460a58f
Boise State	100x	http://coen.boisestate.edu/senocak/files /BSU_CUDA_Res_v5.pdf
Florida Atlantic University	100x	http://portal.acm.org/citation.cfm?id=1730836.1730839& coll=GUIDE&dl=ACM&CFID=88441459&CFTOKEN=90295264
Cambridge University	100x	http://www.wbic.cam.ac.uk/~rea1/research/AIRWC.pdf

State of the Art

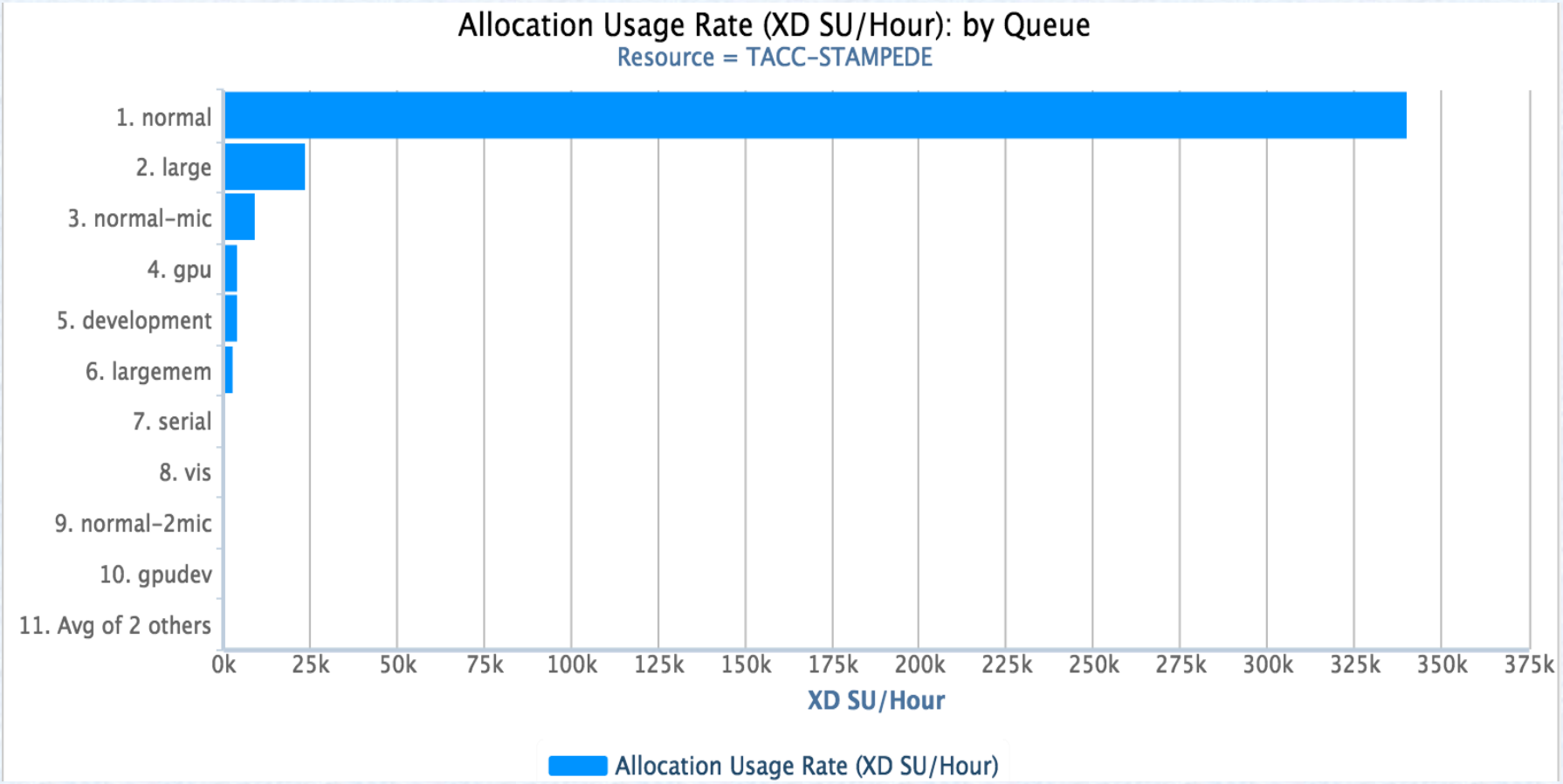
- Accelerated resources widely available (local, campus-wide, national infrastructures like XSEDE)

  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)



State of the Art

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

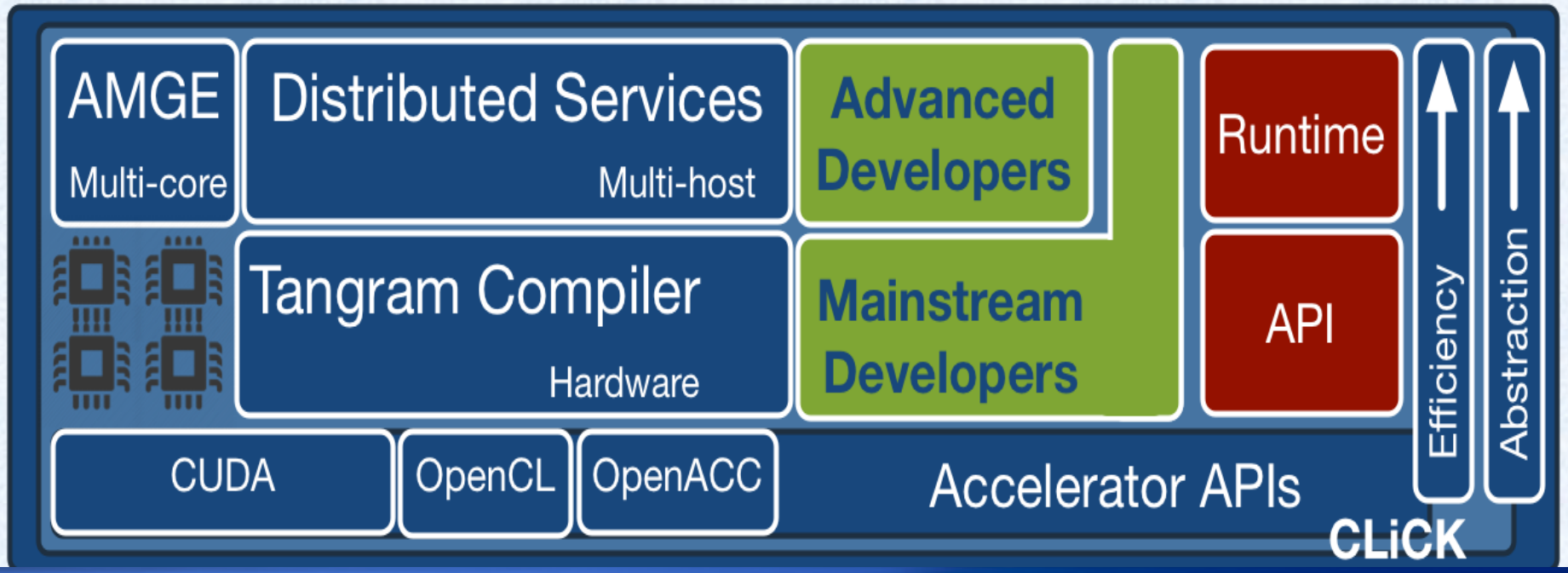
  but...

  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

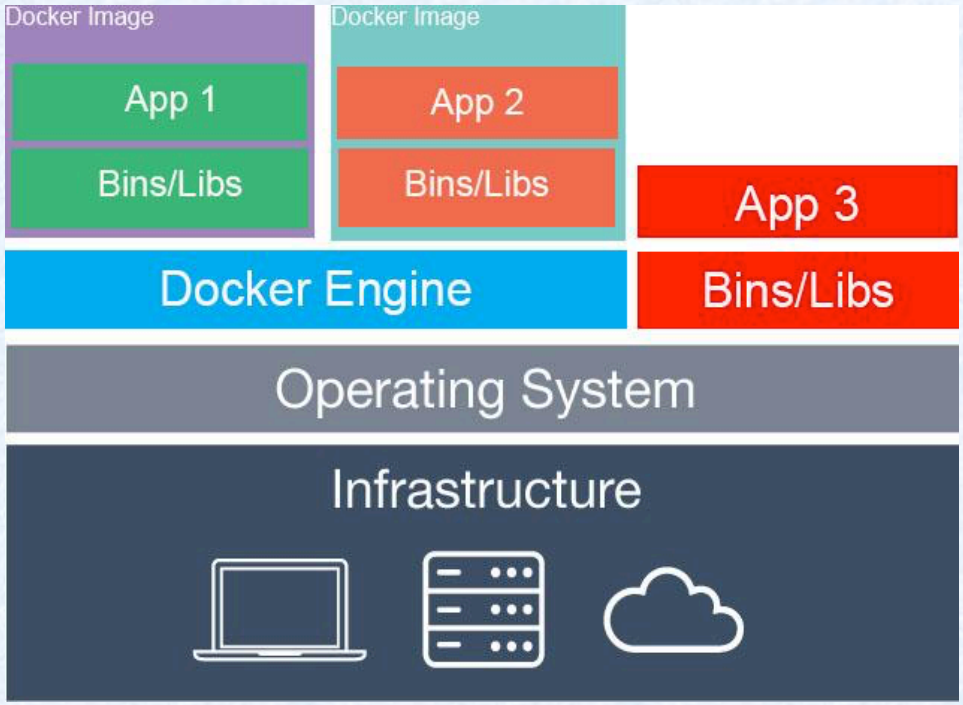


Why Docker?

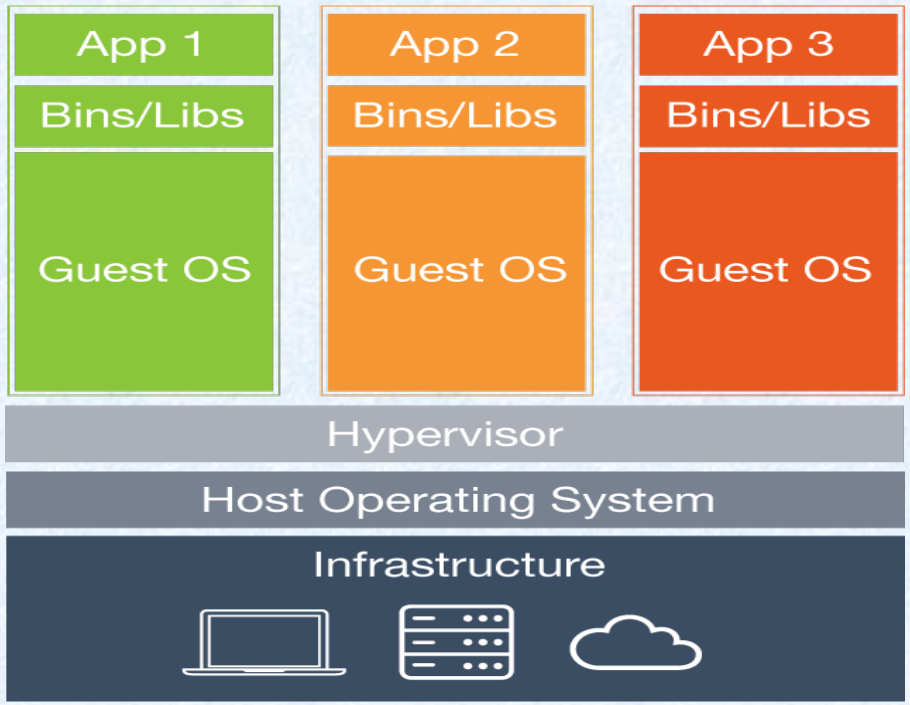
- Virtual images containing only the necessary dependencies and system tools for running applications
- Containers share the kernel of the host operating system resulting in an efficient use of space and computational resources
- The layered filesystem on which Docker images are built allows for multiple 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.

Architecture

- Docker eliminates the need for a hypervisor



Docker

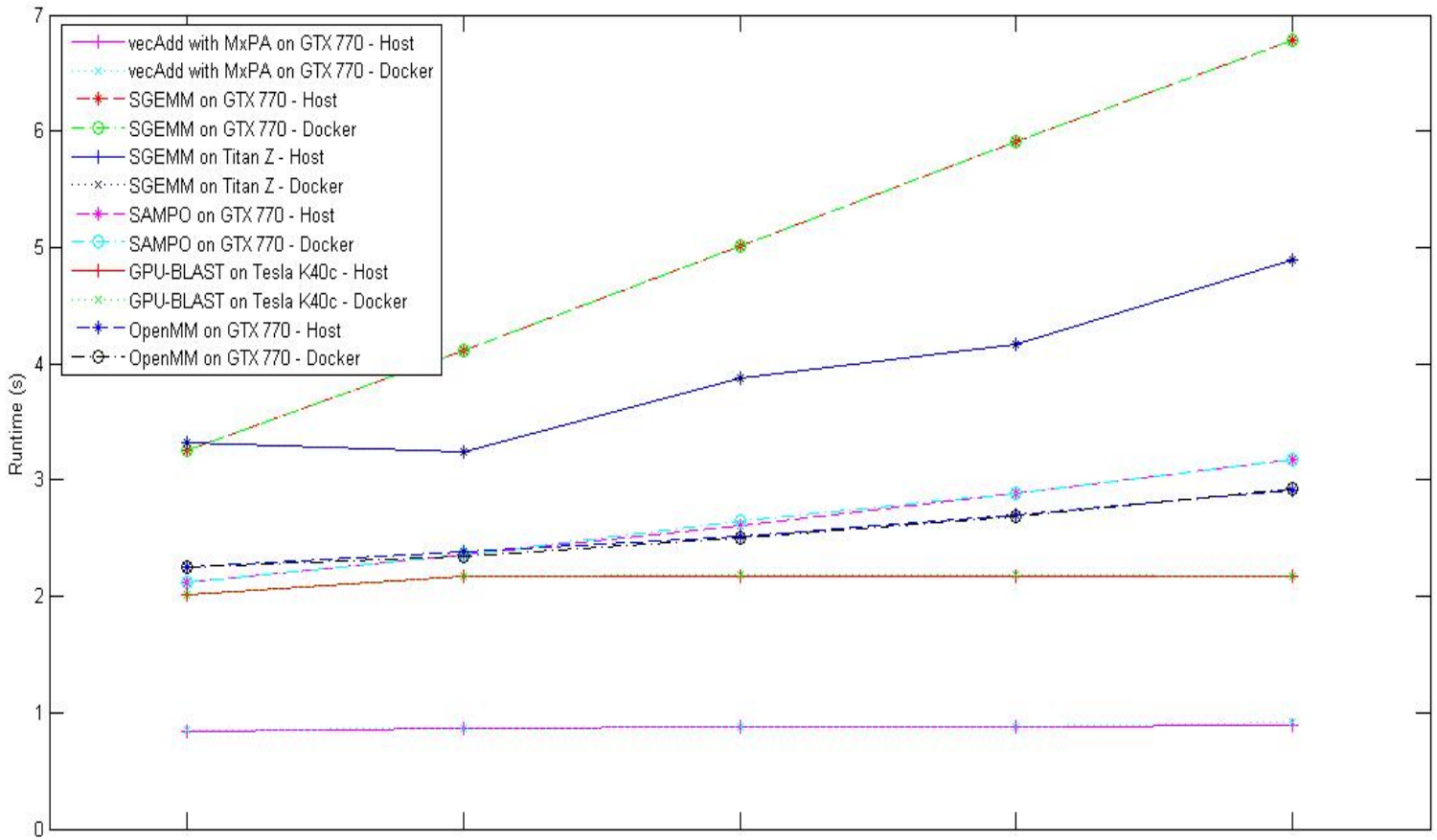


Virtual Machines

Proof-of-concept

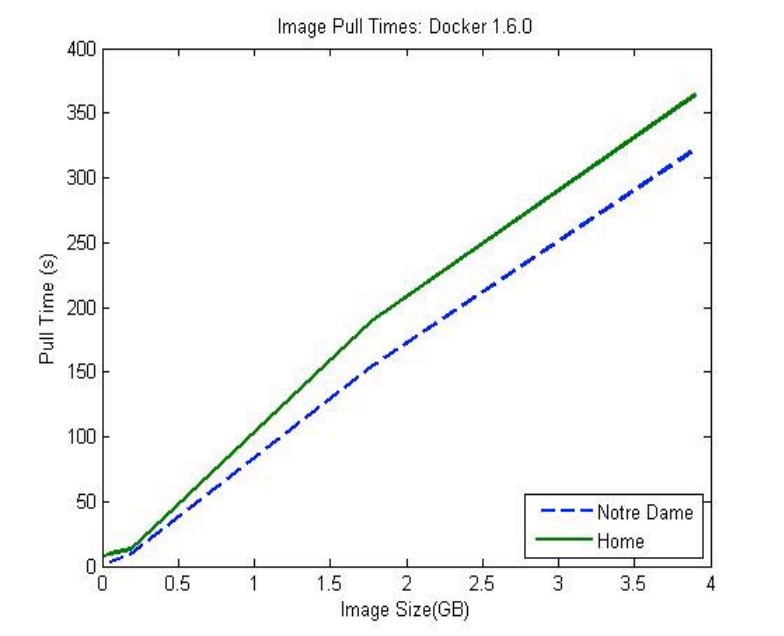
- SAMPO (Scalable Agent-based Mosquito POint model)
  - Models the life cycle of malaria-vectors utilizing the OpenCL API to perform computations on available accelerators (GPU, MICs).
- GPU-BLAST
  - Basic Local Alignment Search Tool developed with CUDA
- SGEMM
  - A common matrix multiplication algorithm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



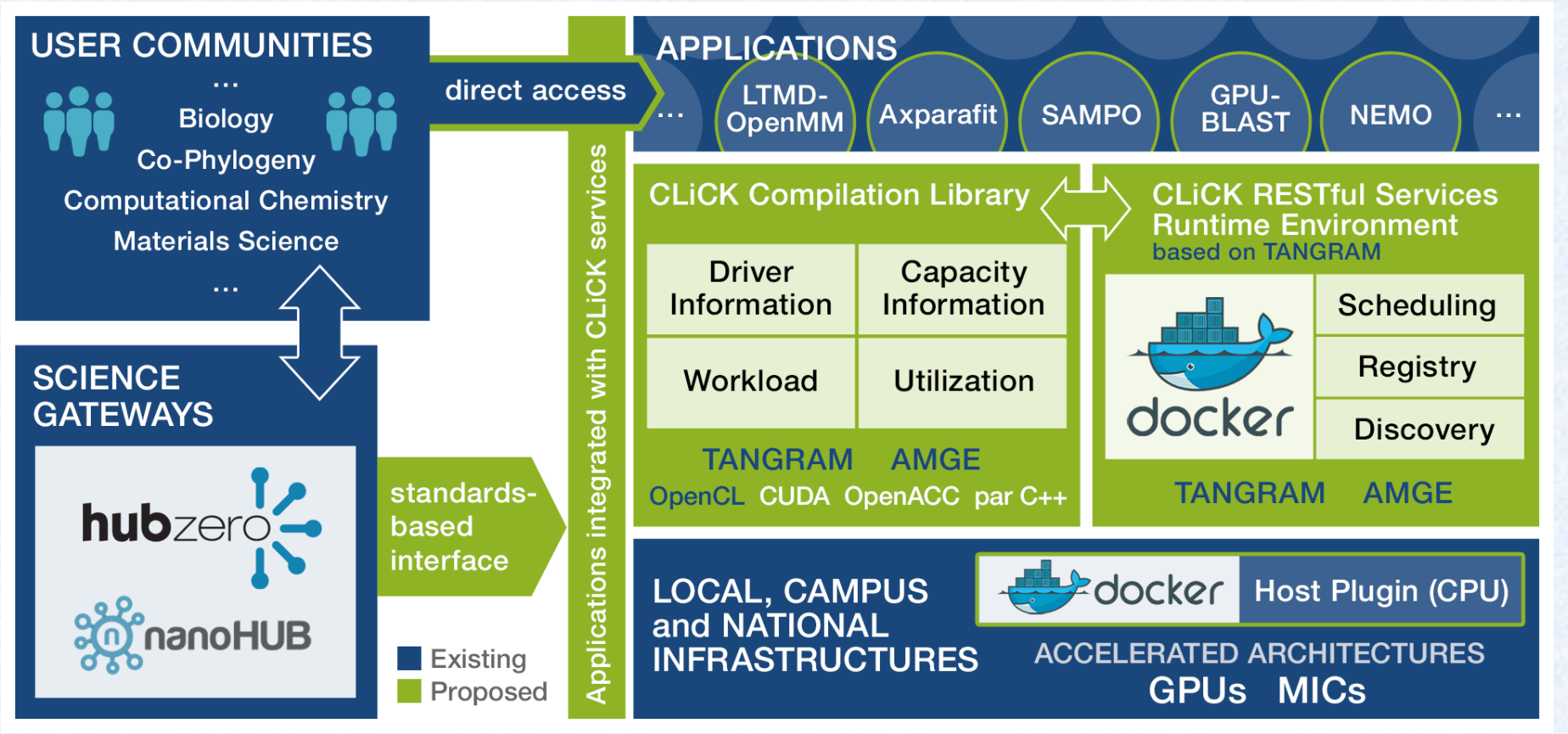
Minimize Transfer Time

- 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

sandra.gesing@nd.edu