



Conclusions

Roberto Gioiosa, Ryan Friese, Alok Kamatar



PNNL is operated by Battelle for the U.S. Department of Energy



Minos Computing Library Tutorial

- **Objectives:** This tutorial provides an overview of the MCL programming environment and a step-by-step guide to write, build, and test an MCL program in a multi-device environment. At the end of this tutorial, attendees should be able to run their MCL code on their laptops and scale out their code on more complex systems, both larger workstations and power-efficient embedded systems.
- **What you have learned:**
 - Develop and build MCL applications
 - Run MCL applications in multi-GPU environment
 - Develop MCL applications that leverage DL accelerators

Resources

- MCL website:
 - <https://minos-computing.github.io>
 - Updates and news
 - Still work-in-progress
- MCL docker container:
 - `docker pull minoscomputing/ppopp21`
 - Playground for MCL
 - Support for NVDLA
- MCL git repositories
 - <https://github.com/pnnl/mcl.git>
 - Also keep an eye on: <https://minos-computing.github.io/github.html>

Beyond MCL...

- The COMET compiler and DSL language for tensor algebra:
 - MLIR-based compiler
 - Generates code for CPUs, GPUs, FPGAs,
 - Runs on CPU parallel, CUDA, Vulkan, OpenCL, and MCL runtime
 - To be release as open source soon (<https://github.com/pnnl/comet.git>)
- The Lamellar distributed runtime
 - Active-message, asynchronous, PGAS runtime for distributed systems
 - Implemented in Rust
 - <https://github.com/pnnl/lamellar.git>

Team



Roberto Gioiosa
PNNL

Roberto.gioiosa@pnnl.gov



Ryan Friese
PNNL

Ryan.friese@pnnl.gov



Alok Kamatar
PNNL

Alok.kamatar@pnnl.gov

FadEzd
]A Eu!c9
iYc7^~ 1
/]Tb
nc-o. ZK
H%mg+
l} ZG
ee q
b@nD?@X
k;nihW
r#P= G u
fUIfa |j
T!)~S
A5vj%z h
<L.N
:dvs ny
o
bī (=x^Rt
f
h 0
c[\n
iK l
f
p ;
?NgE w o
YR:g D
/|~XI
K3k
/go
=



Q&A





Thank you

