Pacific Northwest
NATIONAL LABORATORY

Lamellar:
Rust and HPC

github

docs

crates.io

Ryan D. Friese

Senior Computer Scientist
Future Computing Technologies Group
Pacific Northwest National Lab
ryan.friese@pnnl.gov
509-375-2903

- Principle Investigator for the Lamellar Project
    - Started around spring 2019
- Background in runtime systems,
data movement frameworks, performance modeling,
and HPC Resource allocation

Why Rust?

Rust is a modern systems programing language that is an alternative to C and C++.
It prioritizes reliability, performance, and productivity.

Reliability

- Memory and Thread safety
- Strong compile-time guarantees

Performance

- No garbage collector or runtime
- Zero-cost abstractions

Productivity

- Cargo build tool and package manager
- Verbose and helpful compiler messages

Pacific Northwest
NATIONAL LABORATORY

LAMELLAR: Stack & Nomenclature

github docs crates.io

Lamellar

- A distributed, asynchronous tasking and PGAS runtime for HPC
- **Active Messages**: Distributed async. tasks
- **Darcs**: Dist. Atomic reference counted objects
- **LamellarArrays:** High-level distributed arrays

Thread pool/Async Executor

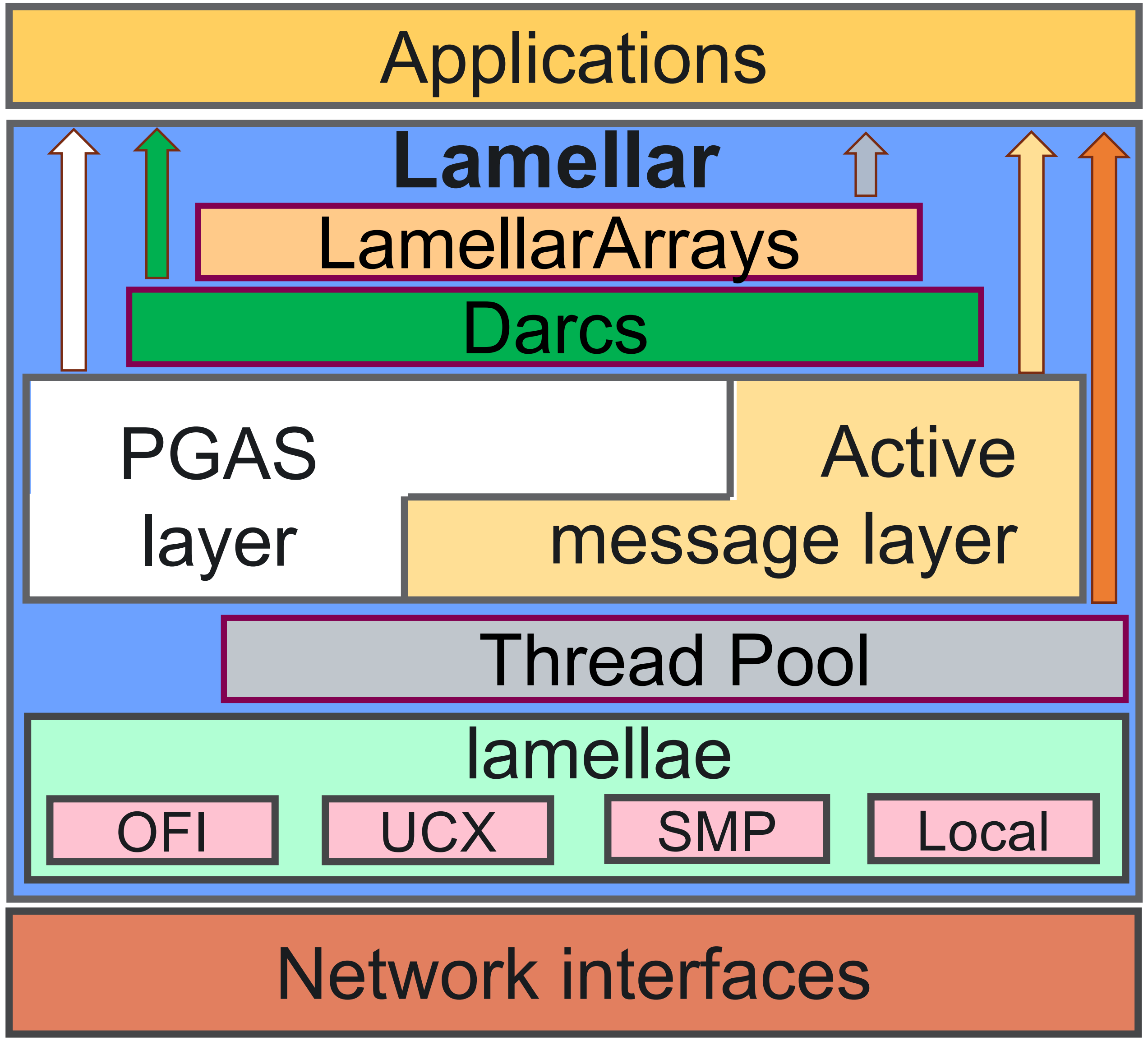
- Active Messages, Futures, Parallel Iteration

Lamellae

- Interface between network devices/drivers and upper-level runtime layers
- Support High Performance Infiniband networks
    - Working on Slingshot support

PE – Processing Element (≅ MPI Rank)

- Multithreaded
- Multiple PEs per hardware node
- (e.g., one per socket)


Applications
Lamellar
LamellarArrays
Darcs
PGAS layer
Active message layer
Thread Pool
lamellae
OFI
UCX
SMP
Local
Network interfaces