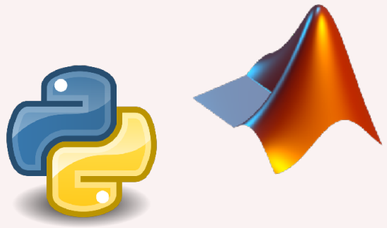
# Introduction to Julia for High-Performance Computing

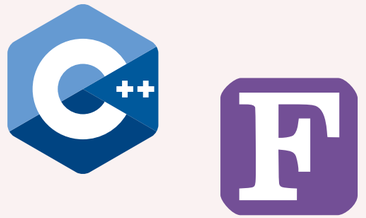
Carsten Bauer @ HLRS, Stuttgart

September 10, 2024

Convenience

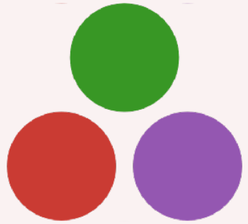


Performance

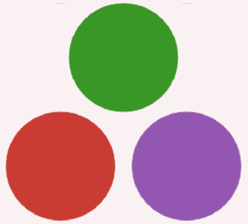


Language Barrier

Convenience



Performance



Gradual transition

| | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|
| | **Foundations** | **Core** | **Node** | **Cluster** |
| 09:00 - 10:45 | Intro<br>Onboarding | Type & Memory Optimizations | Multithreading | Distributed Computing |
| 10:45 - 11:00 | Break | Break | Break | Break |
| 11:00 - 12:30 | Fundamentals | Exercises | Exercises | Exercises |
| 12:30 - 14:00 | Lunch | Lunch | Lunch | Lunch |
| 14:00 - 15:30 | Specialisation & Abstraction | SIMD & Profiling | GPU Computing | Exercises |
| 15:30 - 15:45 | Break | Break | Break | Outro |
| 15:45 - 17:00 | Exercises | Exercises | Exercises | |

# Quick Live Survey

# Julia’s Weaknesses

HPC with Julia is currently a niche.

# Join us at conferences ...




... or in our monthly Zoom call

(open to everyone!)

Achieving
high performance
can be tricky.

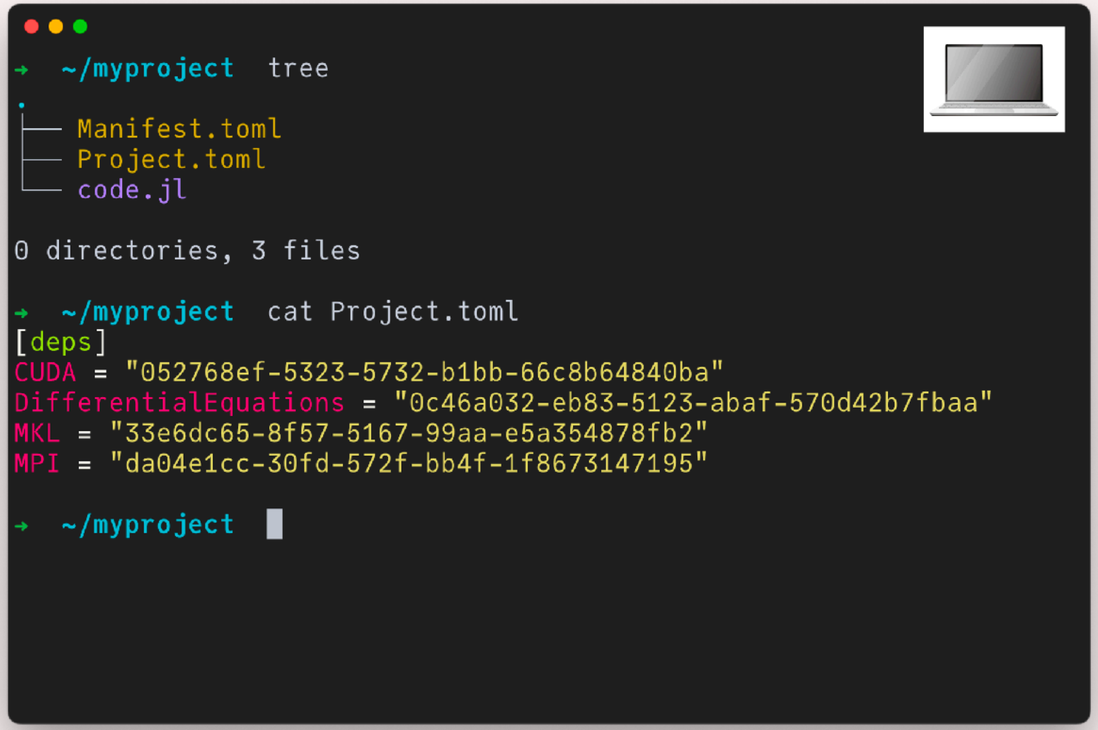
No great way to produce (small) binaries.

# Julia's Strengths

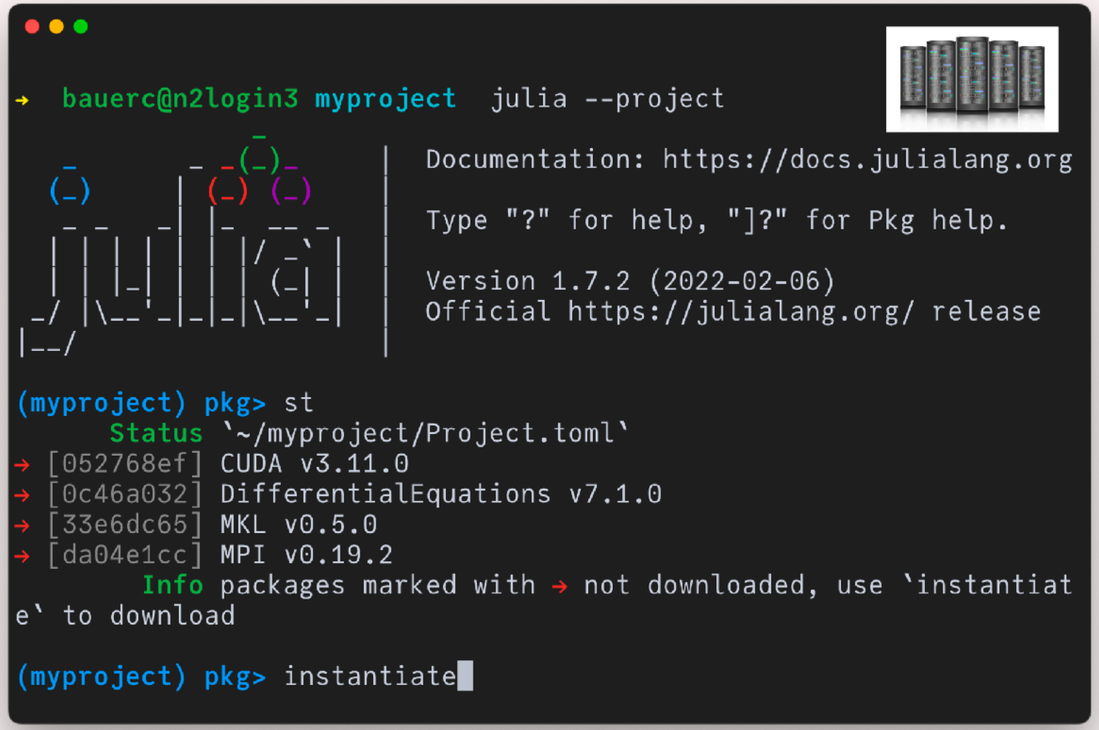
Julia is interactive and convenient.

# Julia has a great package manager

Laptop

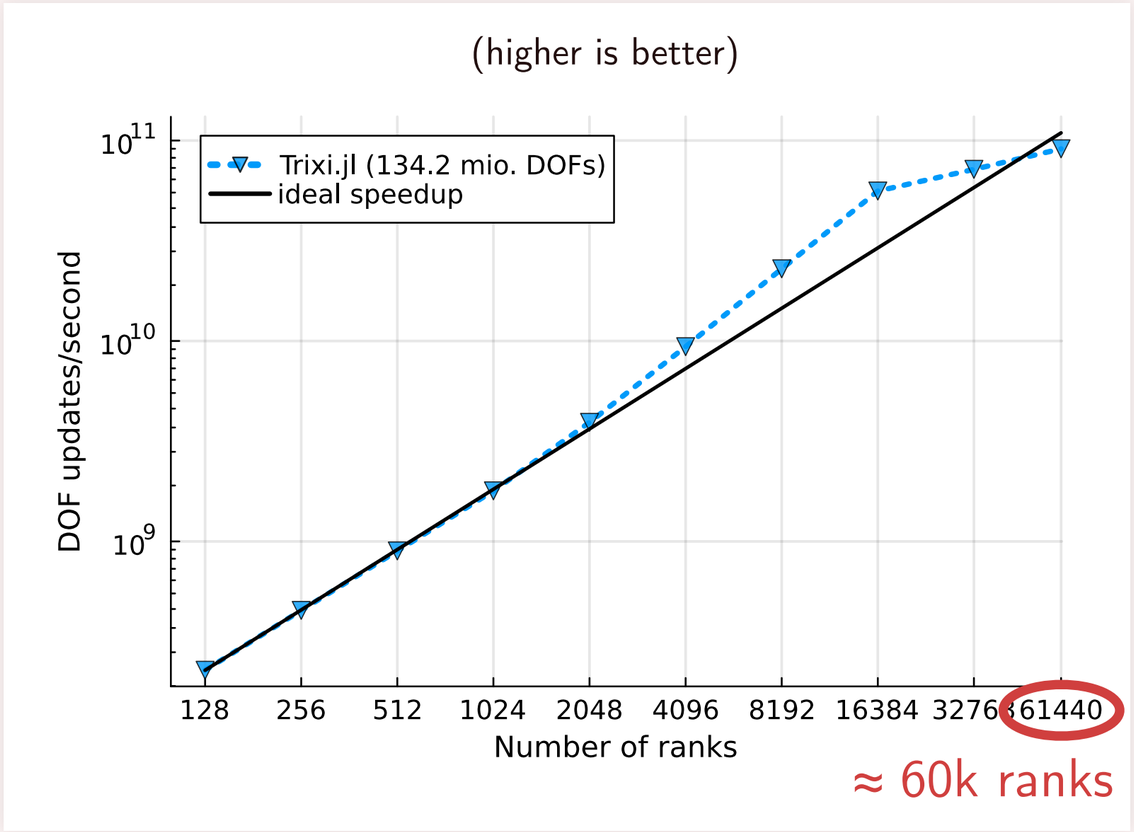



HPC Cluster

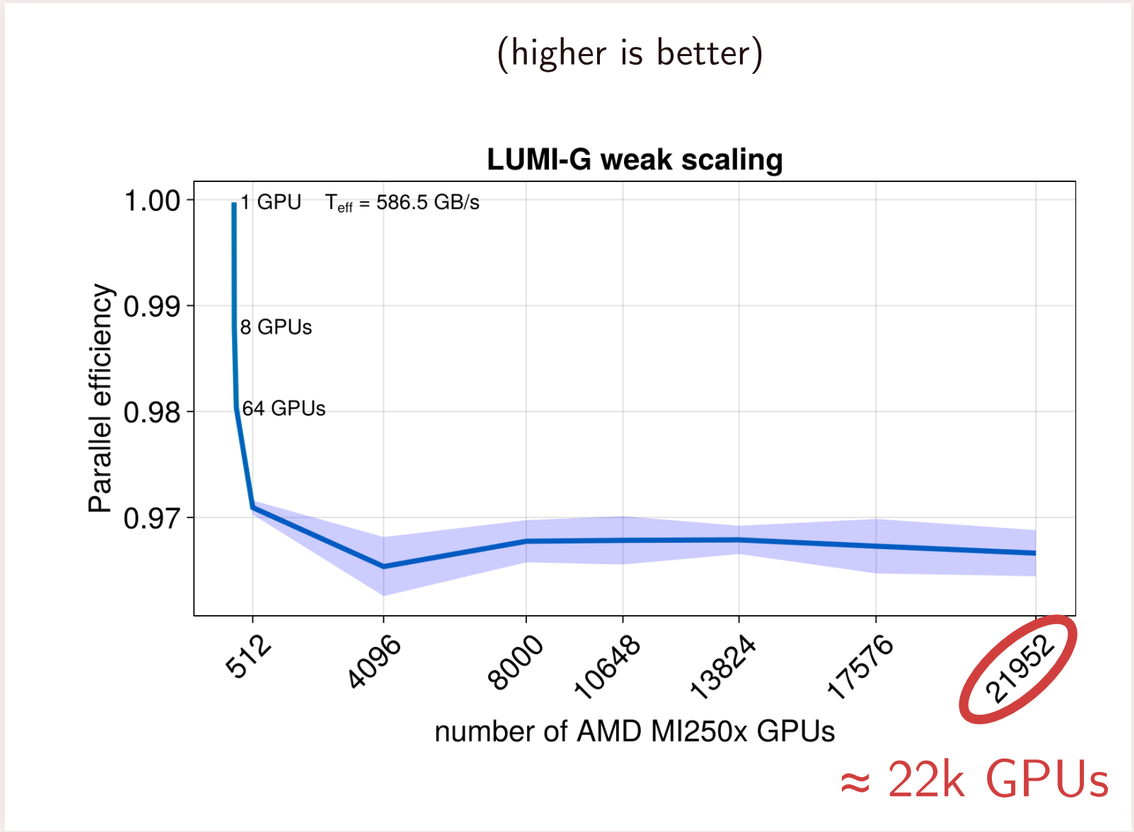


Julia code can be
fast and scalable.

# Example: Good scaling of PDE codes



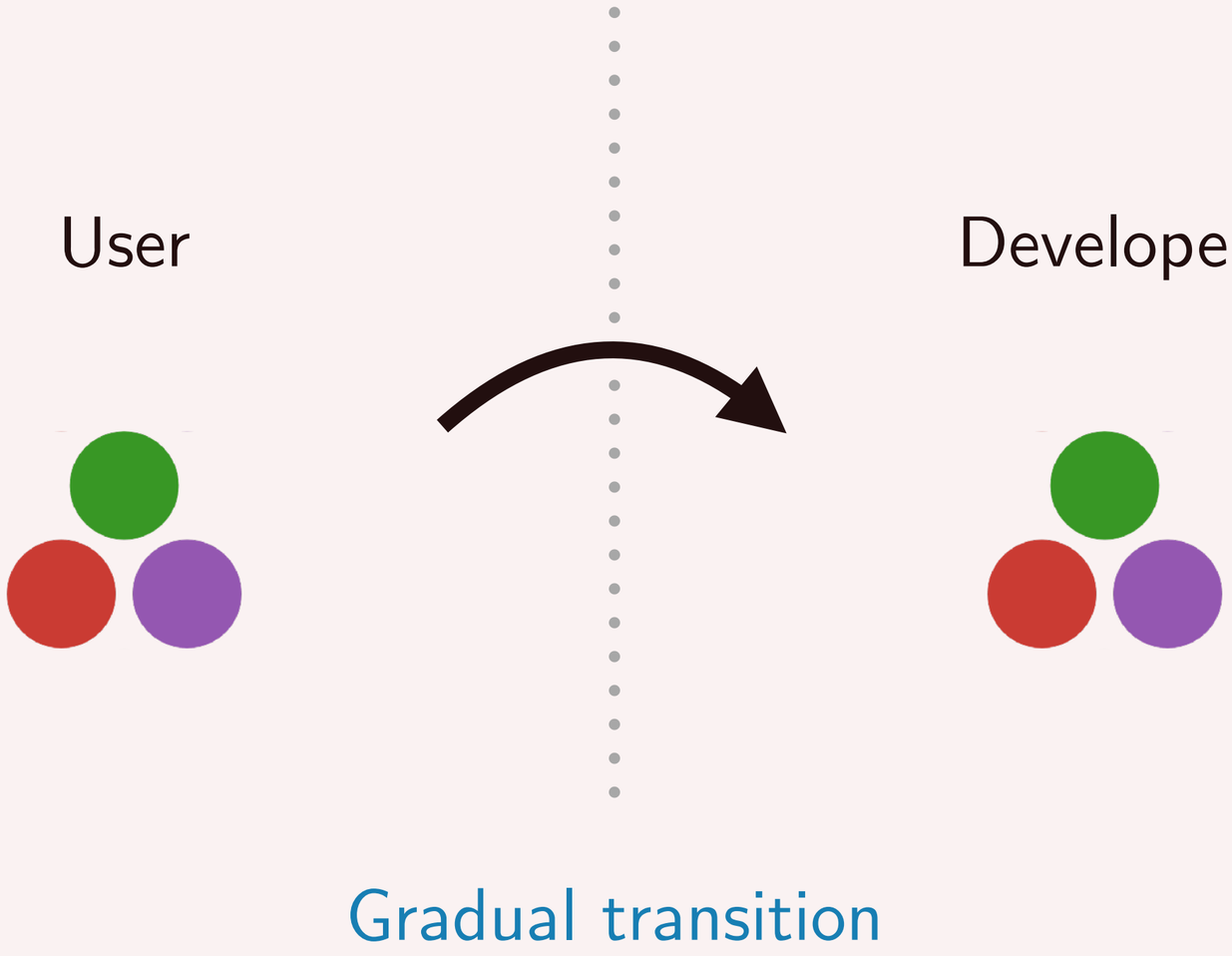

Trixi.jl (Multi-CPU)




ParallelStencil.jl (Multi-GPU)

Julia invites you to gradually delve deeper.

# Julia makes it easier to become a developer.



# Let us delve deeper!