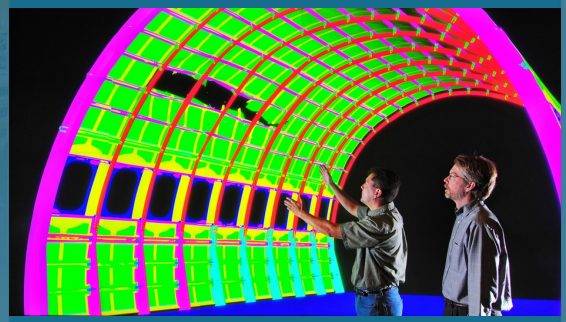
SAND2023-05326C

Sandia National Laboratories

Kokkos 4 – Sustaining Performance Portability for the Exascale Era



kokkos

Christian Trott

Sandia National Laboratories, Center for Computing Research

U.S. DEPARTMENT OF ENERGY | National Nuclear Security Administration

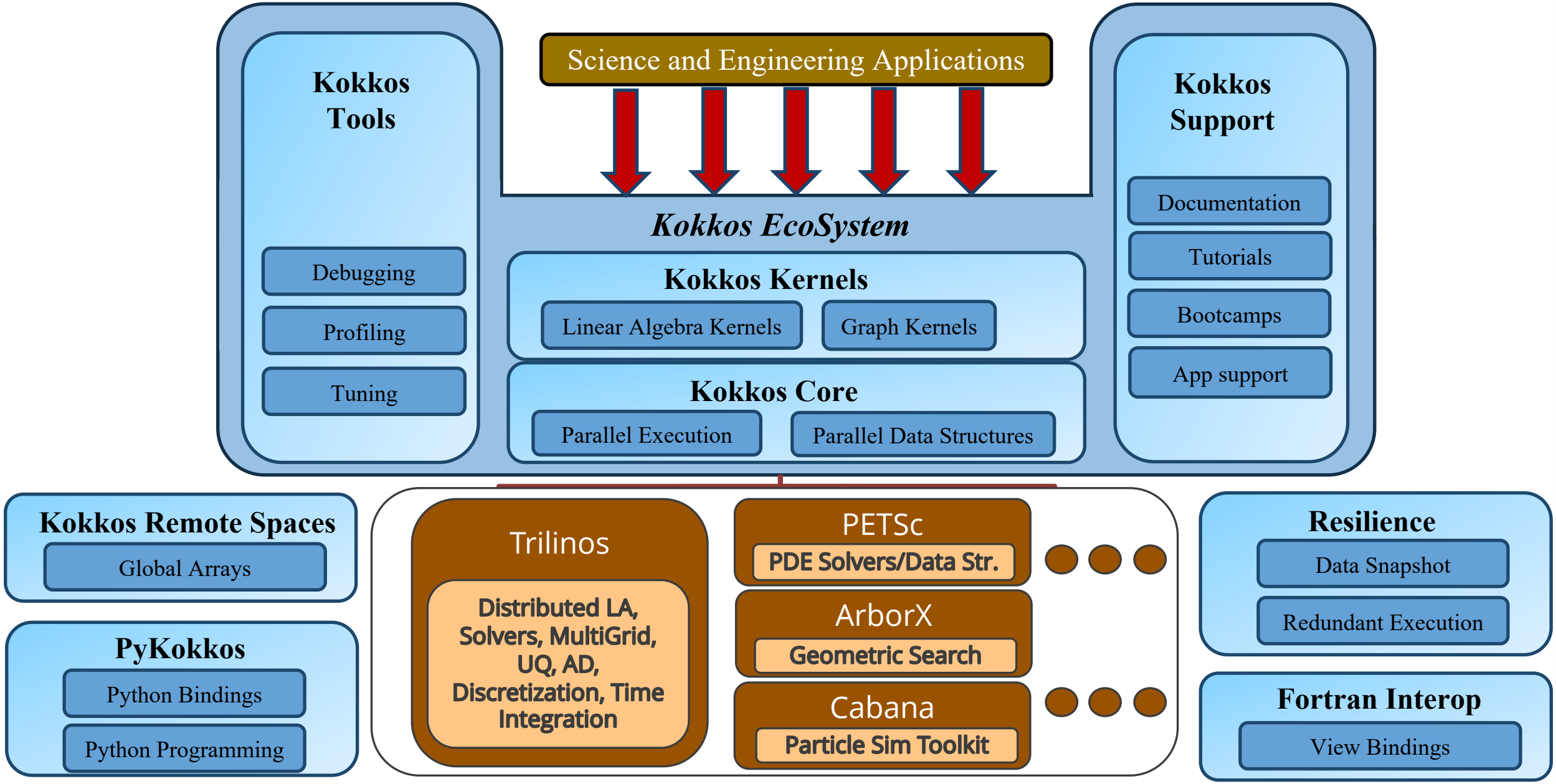
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY | National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The Kokkos EcoSystem - Today

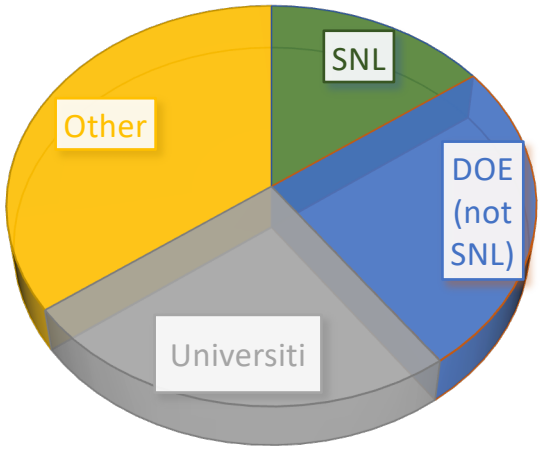


Kokkos Community

Kokkos Slack

https://kokkosteam.slack.com

- >1000 Registered Users
- >150 Institutions
  - Including 34 European



Kokkos Developers

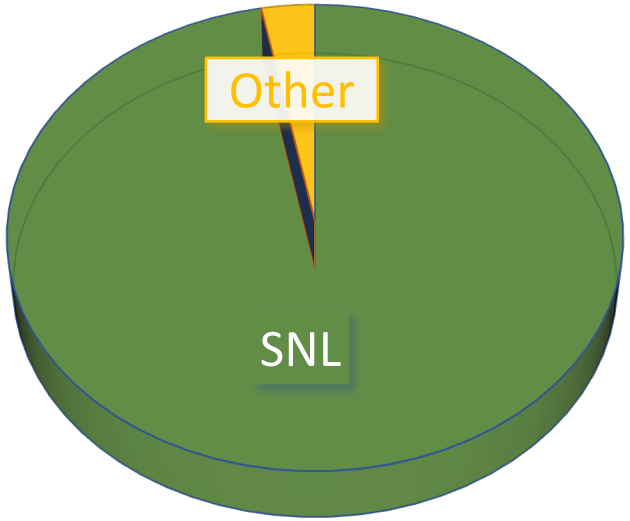
Sandia National Laboratories
OAK RIDGE National Laboratory
Argonne NATIONAL LABORATORY
Los Alamos NATIONAL LABORATORY EST.1943
BERKELEY LAB
CSCS

Applications and Libraries

- Estimated 150-300 HPC projects using Kokkos
- On the order of three-dozen apps run science and engineering production runs with Kokkos
  - Many apps use multiple Kokkos based libraries
- Similar distribution as the Slack User

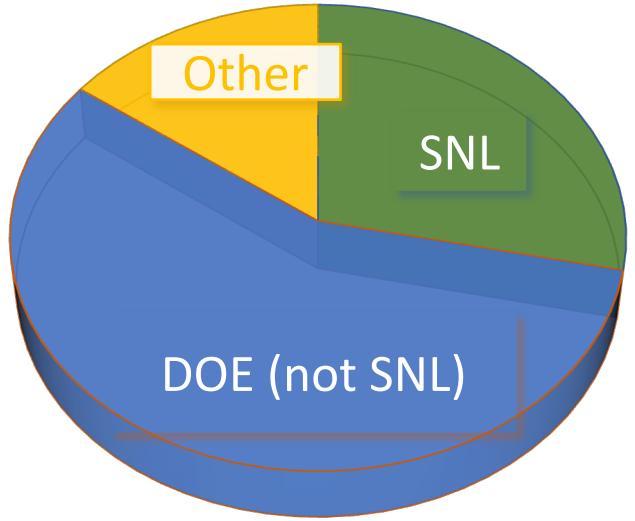
Kokkos Core - Contributions

2015-2017



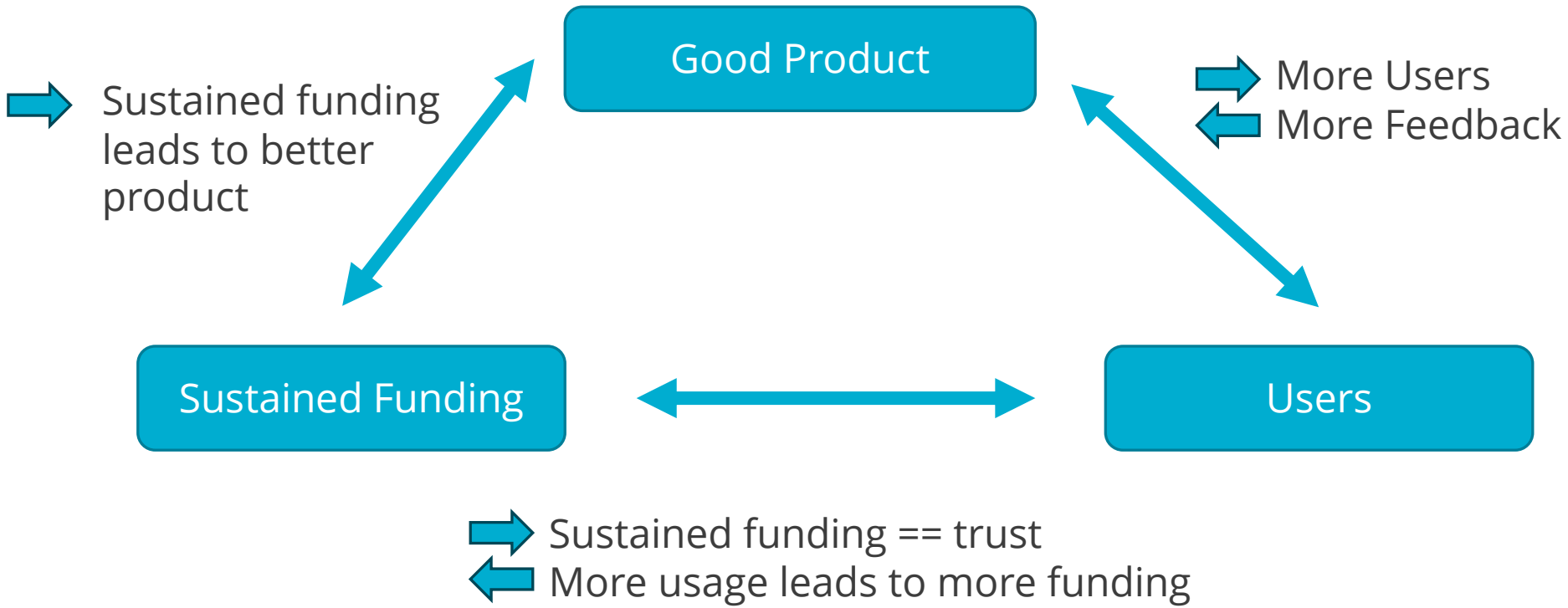
ECP-Funding

2021-2023



- Most of Kokkos-Tools and Kokkos-Kernels development still at Sandia
- ISO C++ Contribution well distributed over labs

Sustainment: A self reinforcing Cycle?



There is strength in numbers: collaboration on core product good for everyone!

Pillars for Long Term Sustainment

Open Source

- Enable wider set of contributor
- Risk mitigation for partner institutions – no one can just take the project away, worst case scenario is institutional fork with internal continued development
- Permissive license critical for industry participation

Core Funding

- Need a group of institutions to sustain core development team
  - NNSA – Sandia National Laboratories *(+ Los Alamos National Laboratory?)*
  - DOE – ASCR Facilities – Oak Ridge National Laboratory, NERSC, … ?
  - CEA

Open Governance

- Encourage participation of institutions by enabling say in direction
  - Enable path for new core funding teams to enter
- Exploring joining Linux Foundation

Kokkos and ISO C++

Long term sustainment via integration of Kokkos features into ISO C++ standard

Getting something into ISO C++

- Requires a lot of effort
  - mdspan was 9 years, but we didn't know what we were doing
  - linalg likely 5 years to get into draft
- Requires prototype and usage experience
  - Need to be able to show successful use in field by sizeable community

Kokkos as the HPCs proving ground

- Large enough community
- More agile development of new features possible
- Kokkos team has gained trust of ISO C++ community as well as standard library implementers

In the standard
- "this" capture C++17
- atomic_ref C++20
- mdspan C++23

In flight for 26
- linalg – BLAS with extensions
- batched_linalg
- mdarray
- submdspan - *approved*
- More accessors and layouts
- simd
- senders/receivers

We need long term engagement with ISO C++ as integral part of Kokkos effort.

My Ideas for Future Directions of Kokkos

Edge computing / Embedded Support

- Many of the same concerns as HPC – resource constraint, performance critical
- Many different devices including FPGAs

Programming Language Safety

- More concern about cyber security – how do we write safer code?
- Kokkos data abstractions (View/mdspan/mdarray) enable safer encapsulation – could make it almost impossible to have out of bounds memory access
- Combined with static analysis could be significant step to enable C++ codes which are memory safe by design

Better integration with distributed computing

- Remote spaces
- MPI interface taking Kokkos data structures


kokkos