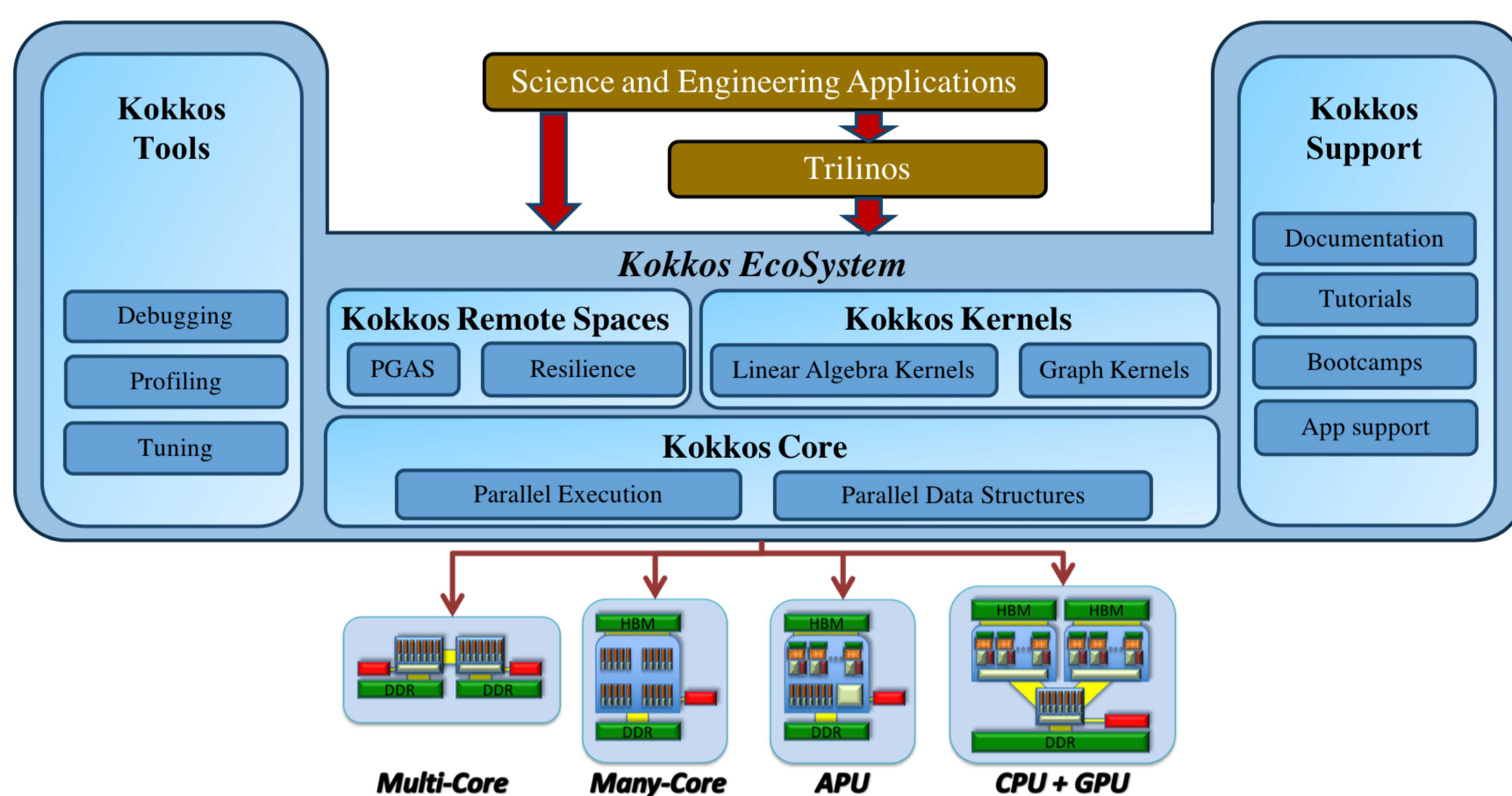


Comprehensive Solutions for Complex Usecases



Support for Users



- Wiki and Website:** Reference Material and Self Study
- Comprehensive API reference à la cppreference.com
 - Over 16 hours of prepared training material

Github Issues: github.com/kokkos

- Tracks all development work
- 353 Issues resolved in FY18 for Kokkos Core



Slack Channel: kokkosteam.slack.com

- Over 10k messages in last 12 months
- Often immediate response, >95% within a day

Bootcamps: Hands on Training and Support for App teams

- Mix of tutorials and app work
- Over 16 hours of prepared training material
- Hands on exercises via Amazon cloud services



ECP Projects Using Kokkos

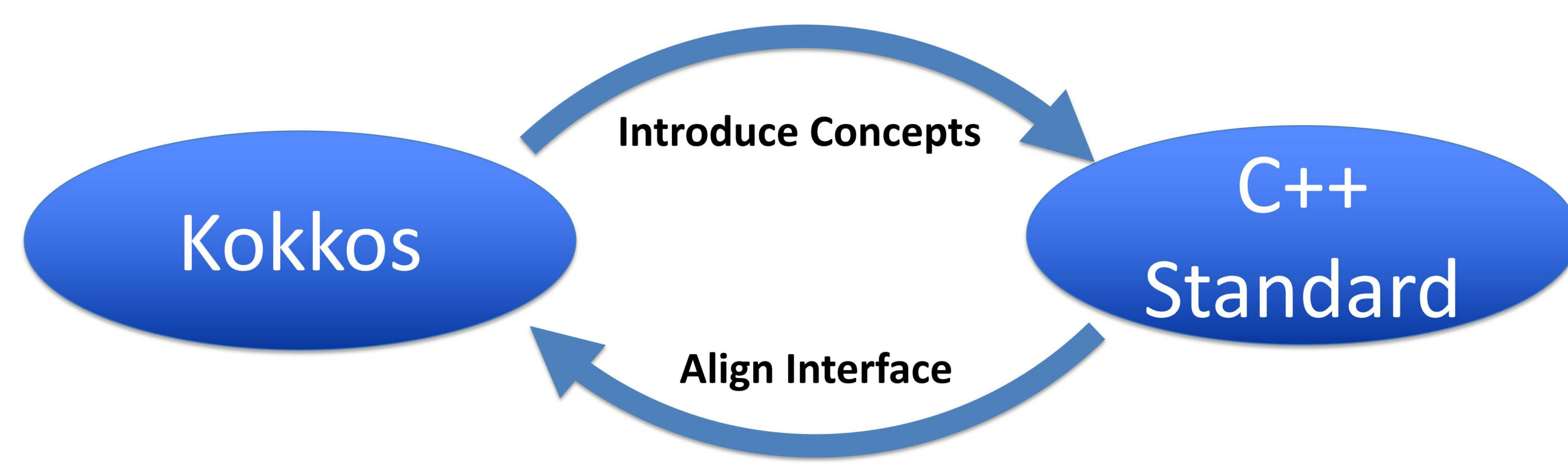
Application	State	Software Technology	State
SNL ATDM Apps	Base Code (SPARC, EMPIRE, Nimble, SPARTA, ...)	SNL ATDM PMR	This is Kokkos ;-)
LANL ATDM Apps	In Parts	LANL ATDM PMR	Evaluation
EXAALT	Base Code	KokkosSupport	
QMCPack	Evaluation	SNL ATDM DevTools	Base Code (in parts)
ExaWind	Base Code	ExaPapi	Integration with KokkosTools
ExaAM	Experimenting	SNL ATDM Math	BaseCode
LatticeQCD	Experimenting	ForTrilinos	BaseCode
ProxyApp	Base Code (in parts)	PEEKs	Base Code
COPA	Base Code		
ExaGraph	Base Code (in parts)		
ExaLearn	Committed (in parts)		

Many other projects in DOE and beyond:

- > 100 projects at least experimented
- If you use Kokkos let us know!

A Vision for the Future:

Kokkos and the C++ standard



Atomic Support in C++20: `atomic_ref`

- Support for any *TriviallyCopyable* type (e.g. `complex<double>`)

Kokkos: `atomic_add(&x(i), 5.0)`
C++20: `atomic_ref(x(i)) += 5.0`

Multi Dimensional Arrays in C++23: `basic_mdspan`

- As `Kokkos::View` it provides multi dimensional arrays with layouts
- Customization point `access_property` allows for extension to support capabilities such as atomic access or restrict
- Subview capability is supported

Kokkos: `View<double*[2][3], LayoutLeft>`
`x("X", N);`
`View<double*[2][3], LayoutLeft, MemoryTraits<Atomic>>`
`x_atomic(x);`
C++23: `basic_mdspan<double, extents<dynamic_extent, 2, 3>, layout_left>`
`x(x_ptr, N);`
`basic_mdspan<double, extents<dynamic_extent, 2, 3>, layout_left, atomic>`
`x_atomic(x);`

Linear Algebra Support:

- Start exploration of adding linear algebra support to the standard
- Goal: Scalar type and layout agnostic via `mdspan` as basic data structure

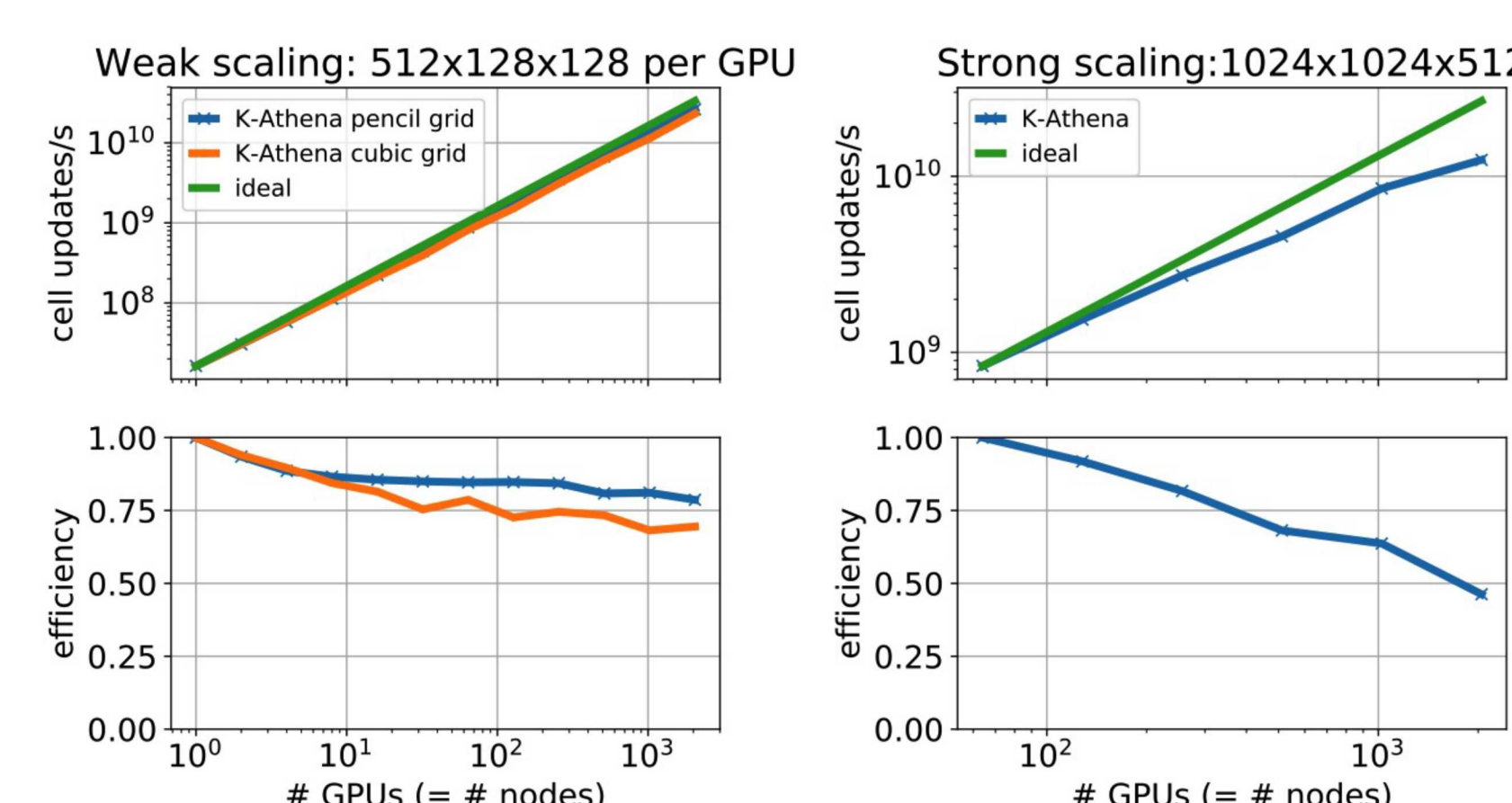
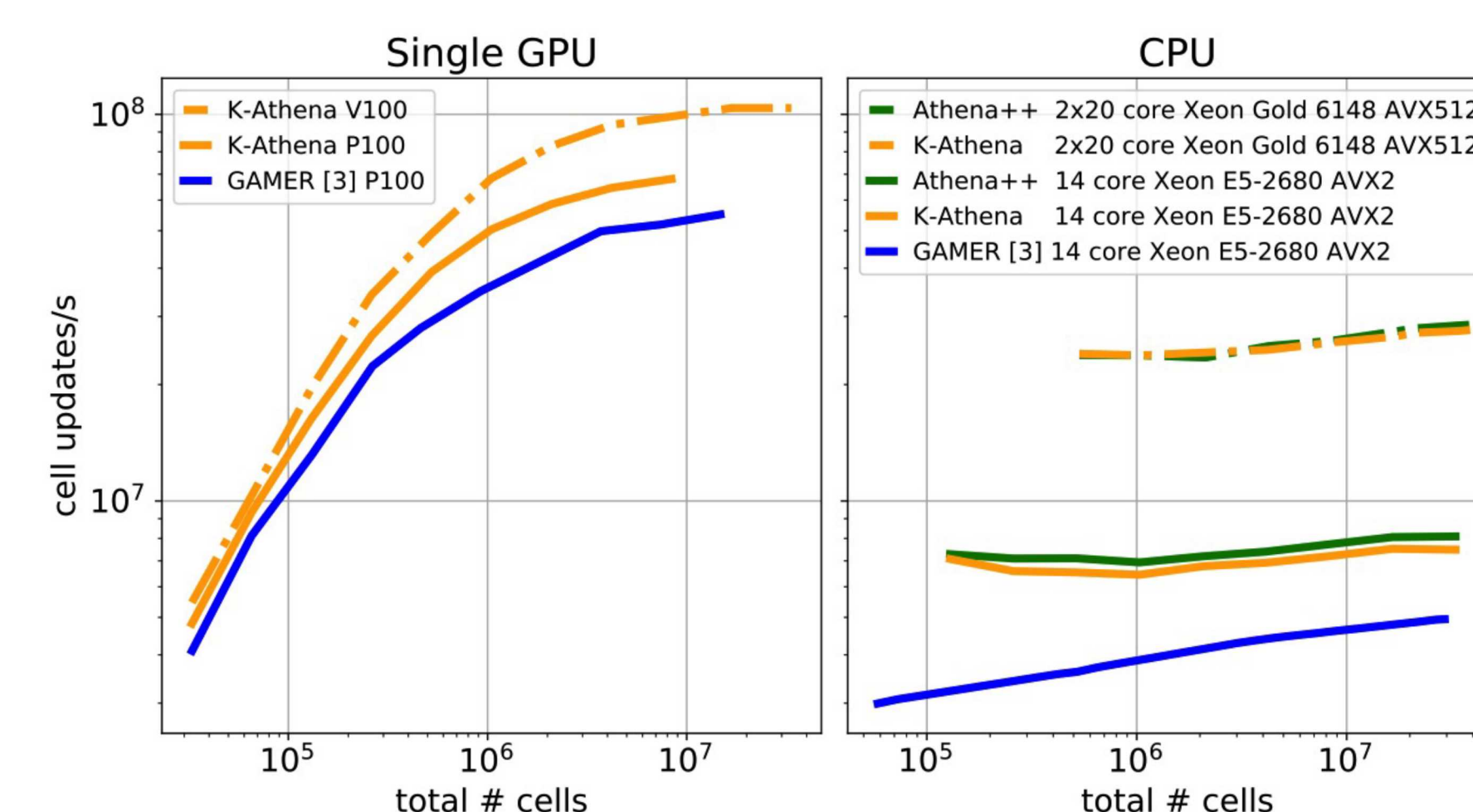
Bootcamp Case Study: K-Athena

Porting Experience from Bootcamp + Continued work after that

- Based on Athena++'s adiabatic ideal MHD solver
- Finite Volume astrophysics MHD code
- Athena++ only works on CPUs
- K-Athena developed by F. Glines, P. Grete and B. O'Shea from Michigan State University

Single Node Performance

- GAMER is a similar code, same algorithms with CUDA support
- K-Athena beats CUDA code while being code to CPU optimized code



Plots courtesy of F. Glines, P. Grete and B. O'Shea

Scaling on Titan

- K-Athena scales well with weak scaling and strong scaling to 1000s of GPUs

Upcoming Events

BootCamp @ NERSC: March 2019

- Basic Kokkos Training
- Hands-On Apps work

1st Kokkos Usergroup Meeting @ Sandia: April 2019

- Application talks with a focus on challenges and solution strategies
- Roadmap discussions
- Area focus sessions

BootCamp @ East-Coast: Summer 2019

- Basic Kokkos Training
- Hands-On Apps work