

# Evaluating Production Engineering Application Performance on the NNSA Trinity Advanced Technology System

Sandia National Laboratories, NM

Courtenay Vaughan, Dennis Dinger, Paul Lin, Simon D. Hammond, Doug Pase, Christian R. Trott, Robert E. Benner, Jeanine Cook and Robert J. Hoekstra

## NNSA/ASC Trinity ATS Platform

- Cray XC40/50 Platform with the high-performance Cray Aries DragonFly interconnect
- Architected by Los Alamos and Sandia National Laboratories for the production NNSA/ASC capability computing campaigns
- **Split into two phases of deployment:**
  - **Phase I (circa 2015)** – approx. 9,000 nodes of dual socket, 16-core Intel Haswell E5 series Xeon processors
  - **Phase II (circa 2017)** – approx. 9,000 nodes of 68-core self-booting single socket Intel Xeon Phi (Knights Landing) processors



## Kokkos Profiling Hooks and Tools

### User Application:

```
void mykernel() {
    ..
    Kokkos::parallel_for (N, KOKKOS_LAMBDA (const int i) {
        printf ("Hello from i = %i\n", i);
    });
    ..
}
```

### Profiling Tool:

```
START_KERNEL_PROFILER
(e.g. START_TIMER)

END_KERNEL_PROFILER
(e.g. END_TIMER)
```

### Kokkos Runtime:

```
Kokkos::parallel_for (...) {
    CALL_PROFILE_HOOK_START

    PARALLEL BODY

    CALL_PROFILE_HOOK_END
}
```

- Dynamically loaded at initialization and built directly into Kokkos runtime by default (no special linking, no LD\_PRELOAD)
- Standard tools include kernel timers, memory/data structure profiling, cache miss/hit rates, memory bandwidth *etc*
- Can be used to directly connect into vendor tools (e.g. Intel VTune, NVIDIA NSight, CrayPAT, IBM Performance Toolkit *etc*)
- Working with several DOE-ECP projects to extend profiling capabilities into new tools

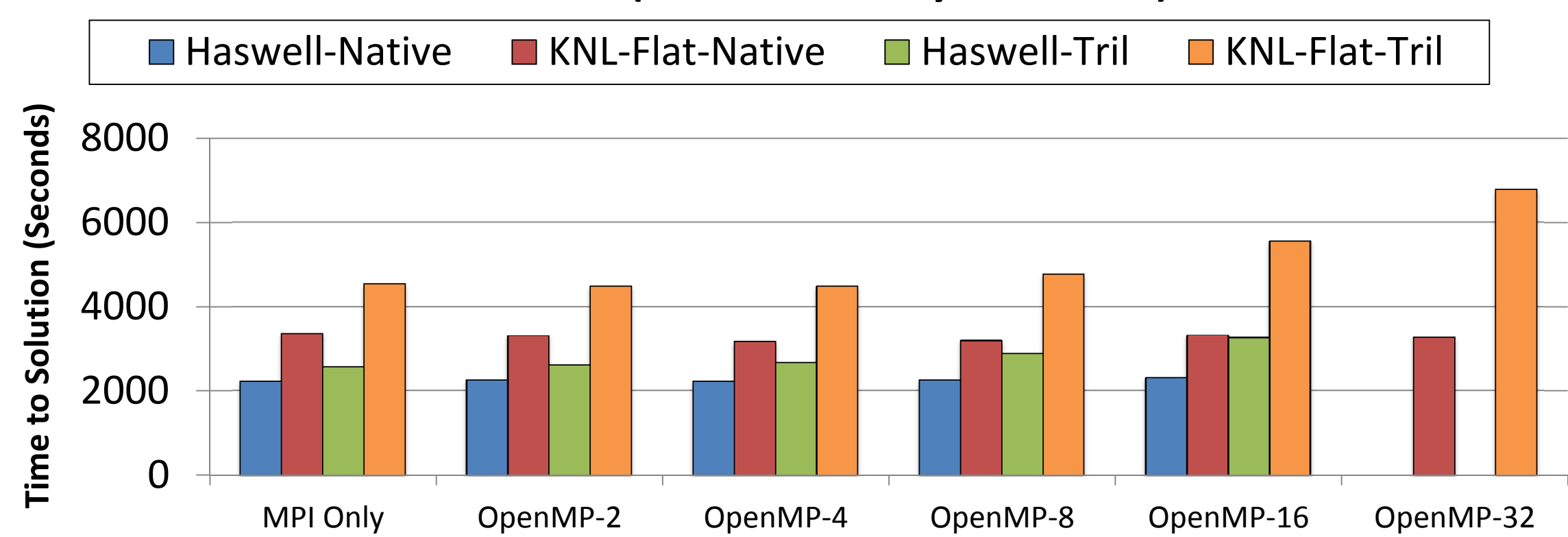
- <http://www.github.com/kokkos/kokkos>
- <http://www.github.com/kokkos/kokkos-tools>

## ATDM SPARC Thread Scaling Analysis

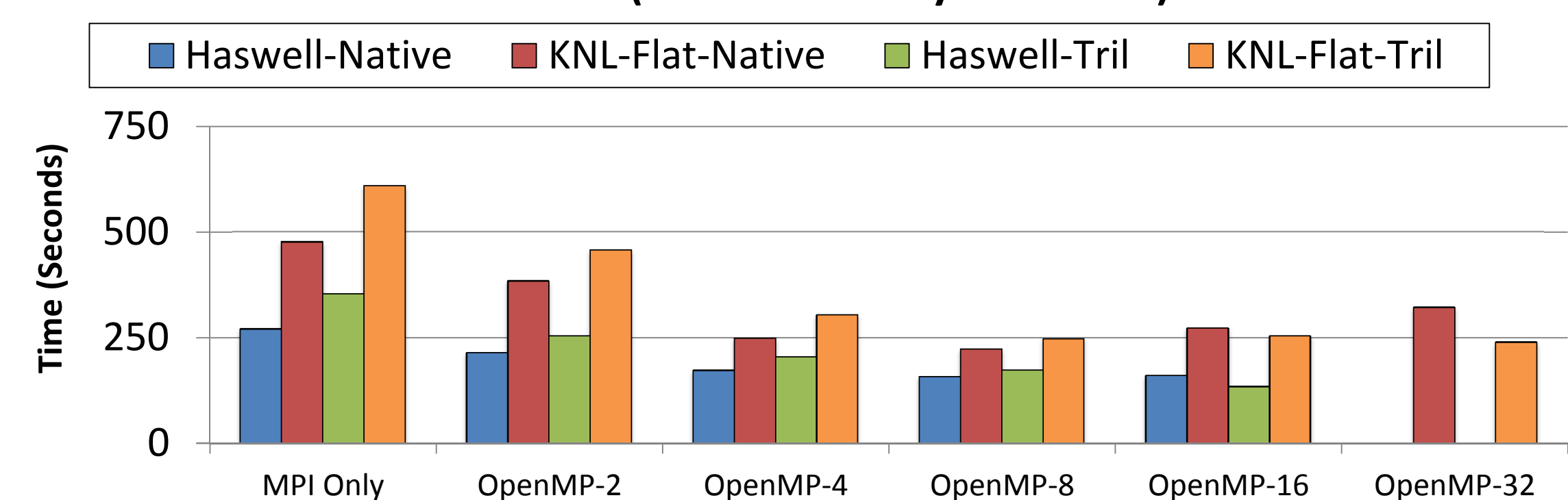
ATDM-SPARC application is a hypersonic CFD modeling application developed from the ground up to be on-node parallel using the Kokkos C++ performance portability abstraction.

- Runs on Intel Xeon, Knights Landing, IBM POWER8 and NVIDIA GPUs from same source
- Can switch between “native” (in-house optimized) and Trilinos solvers
- Utilizes MPI communication between nodes but experimental work on multi-node tasking runtimes in development

**SPARC Time to Solution on Trinity Phase-I and Phase-II Processors (64 Node Test System Runs)**



**Time Spent in MPI Operations on Trinity Phase-I and Phase-II Processors (64 Node Test System Runs)**



## ATDM SPARC Kernel Analysis (Knights Landing)

