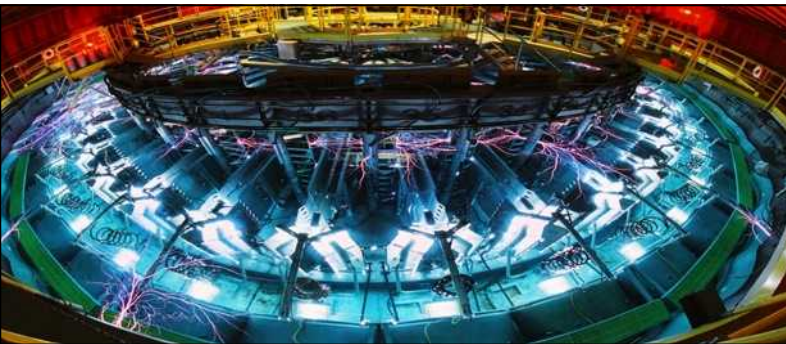


Exceptional service in the national interest



KokkosP: Runtime Hooks for Portable Performance Analysis

S.D. Hammond, C.R. Trott, H.C. Edwards, N. Ellingwood

Center for Computing Research,

Scalable Computer Architectures

Sandia National Laboratories, NM

sdhammo@sandia.gov



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Our Research Vision

“..we should have a **single application code base** which runs across multiple platforms and architectures *and* achieves as **close to 90% of the native/best programming model/solution** as possible.”

- This is *really* hard
- But .. its what our code teams expect (and deserve)
- Requires us not just to be content to write Kokkos but also to write solutions in other models to assess how well we are doing
- We have learned a huge amount about Kokkos (and other models) this way

Performance Tools

- One of the problems with this approach has been that we find performance differences
- Then .. we need code ninjas to work out where some of the time is going
- Existing performance tools have some real weaknesses
 - They don't understand our abstractions
 - Some don't have a one-to-one mapping
 - Most are single-type of device
 - Non-uniformity of tools across architectures



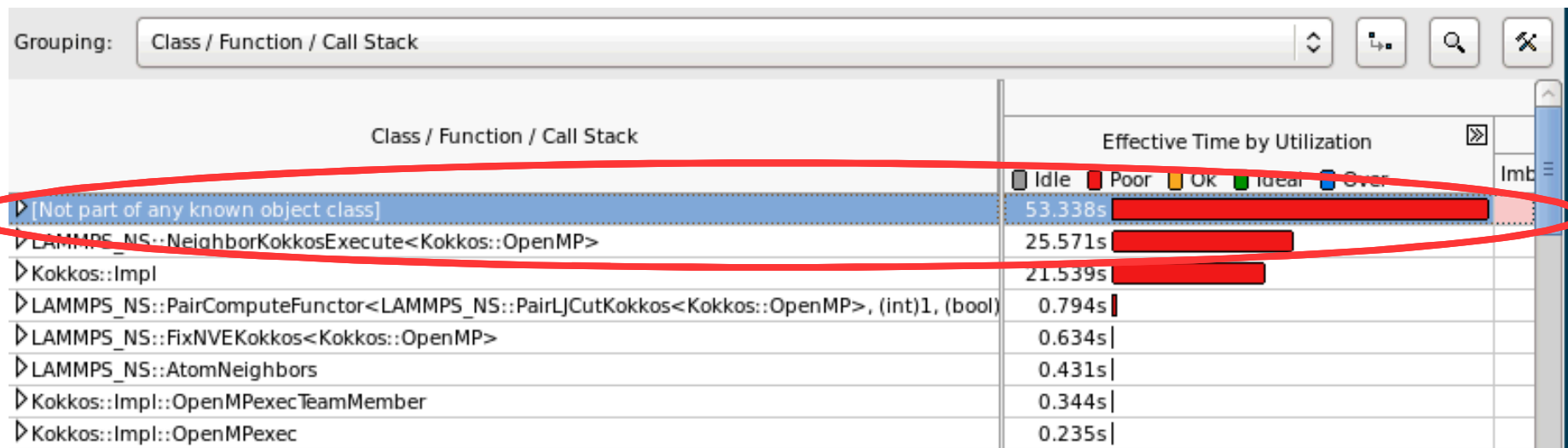
KokkosP Hooks

- Integrated hooks directly into the Kokkos runtime
 - Parallel kernel dispatch
 - Memory allocation using Kokkos views
 - User-defined section naming
- Dynamically loaded at runtime (no LD_PRELOAD)
- Stackable (one tool calls another, calls another)
 - Record multiple items of interest in a single run
 - Easily filter
- Working on dynamic tool feedback API
 - Control threads, data placement etc



Profiling Kokkos with VTune

- Programming abstractions obscure the call stacks
- Confusing identification of Parallel Regions
 - OpenMP parallel for is in a single file: Kokkos_OpenMP_Parallel.hpp
- Very long function names which users absolutely hate



Profiling Kokkos: KokkosP+VTune

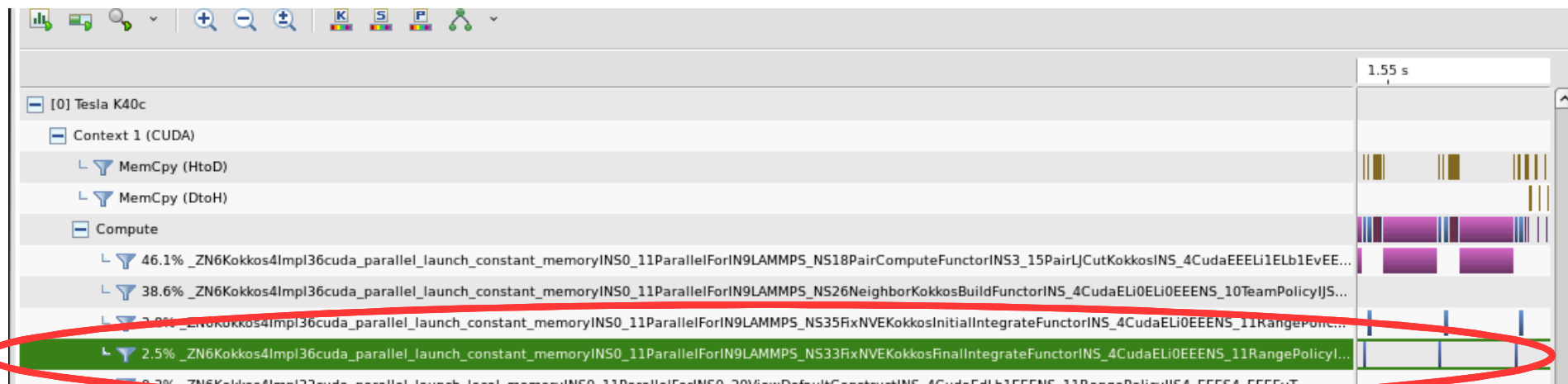
- KokkosP tools add domain marking for kernels
- VTune allows filtering, zoom in, etc. based on Domain and Frames
- Track memory allocations in timeline viewer
- Domain markings can also make CUDA kernel execution visible

Grouping: Frame Domain / Frame / Function / Call Stack

Frame Domain / Frame / Function / Call Stack	CPU Time								
	Effective Time by Utilization			Spin Time			Overhead Time		
	Idle	Poor	Ok	Imbalanc...	Lock ...	Other	Creation (OpenMP)	Sch...	Red...
ParallelReduce.PairForce	42.993s			3.226s	0s	0s	0.011s	0s	0s
ParallelFor.NEigh::Build	35.089s			0.798s	0s	0.000s	0.004s	0s	0s
ParallelFor.22BuildBorderList	22.546s			0.010s	0.001s	-0.00 ...	0.000s	0s	0s
[No frame domain - Outside a	0.204s			2.250s	0.011s	0.005s	0.022s	0s	0s
ParallelFor.N6Kokkos4Impl20	0.636s			0.271s	0s	0.000s	0.027s	0s	0s
ParallelFor.N6Kokkos4Impl20	0.016s			0.260s	0s	0s	0.602s	0s	0s
ParallelFor.NVE::final	0.509s			0.107s	0s	0s	0.002s	0s	0s

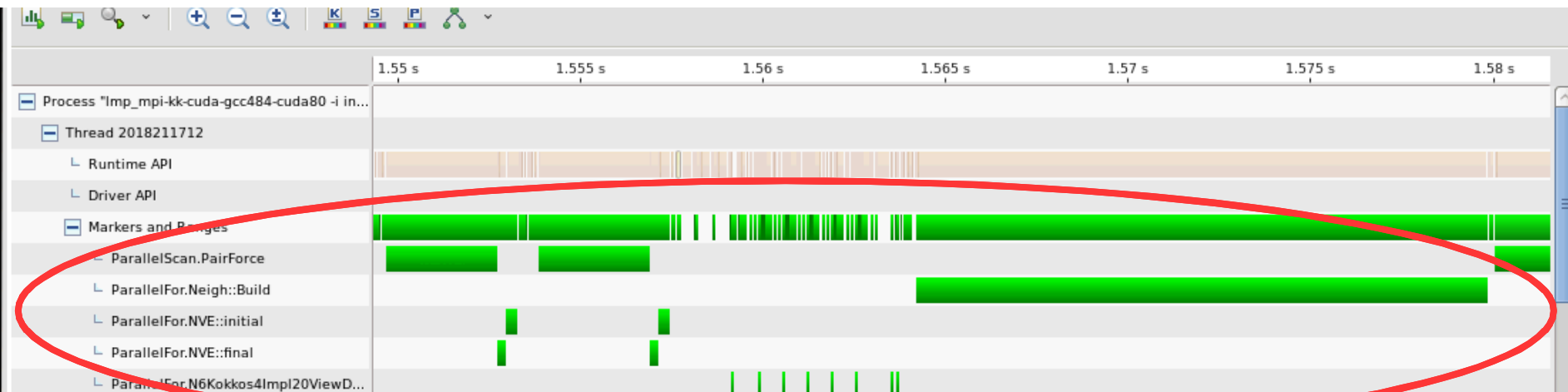
Profiling Kokkos: NSight

- NSight critical for performance optimization on GPUs
 - Bandwidth analysis
 - Memory access patterns
 - Stall reasons
- **Problem:** again template based abstraction layers make awful function names, even worse than in VTune for some problems



Profiling Kokkos: KokkosP+ NSight

- KokkosP uses the CUDA8-NVTX Interface
 - Named Domains in addition to named Ranges
 - Using KokkosP NVProf-Connector to pass user-provided names through
 - Shows Host Regions + GPU Regions



Summary

- Just a snapshot of the tools interfaces and *connectors* we have been working on
 - Connections for parallel execution patterns
 - Memory allocations
 - User-defined named regions
- Simple tools require no vendor products (all in-house)
- But .. users can also use their favorite vendor tools (VTune, Nsight, etc)
- Research prototypes:
 - Feedback API (tools dynamically feed info back to application/Kokkos)
 - Debugging connectors
 - Vectorization and Instruction Analysis (APEX LDRD at Sandia)
- <http://www.github.com/kokkos>



**Sandia
National
Laboratories**

Exceptional service in the national interest

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