Kokkos Strategic Perspective

Applied Computer Science Meeting
Livermore, CA
February 8, 2017
SAND2017-1198 PE
What is Kokkos?

- An answer to: What is “X” in ”MPI + X” ?
  - On-node / shared memory parallelism for next generation platforms (NGP)
    - Programming model
    - Library implementation on CPU, GPU, MIC, ... architectures
  - For applications and domain libraries written in C++11 (or later)
  - Providing performance portability and productivity

- An open source software product
  - https://github.com/kokkos
  - BSD 3-clause license

- A Greek word: ΚÓΚΚΟΣ
  - Translation: “granule” or “grain” ; like grains of sand on a beach
  - Not an acronym
What is Kokkos?

Kokkos: performance portability for C++ applications

Applications & Libraries

LAMMPS
Albany
SIERRA

EMPRESS
SPARC

Trilinos

Multi-Core

Many-Core

APU

CPU+GPU
Where did Kokkos come from?

- **2003-2010 Kokkos V1 (retired in 2014)**
  - Trilinos’ on-node sparse linear algebra kernels with portable C++ API
  - V1 Copyright in 2008
  - Funding: DOE ASC, DOE ASCR, SNL LDRD

- **2010-2017 Kokkos V2**
  - Performance portable programming model and C++ library implementation
  - V2 Copyright in 2014 (V3 copyright planned for 2017)
  - Move to github 2015, independent from Trilinos
  - Funding: DOE CSRF, DOE ASC, DOE ECP, SNL LDRD
Where is Kokkos now?

- **Research and Development – DOE ASC/ATDM**
  - Core team @ SNL, *and* we accept github pull requests
  - New and improving capabilities and architecture back-ends

- **Software Configuration Management – via github.com**
  - Version control: *master* release branch, development branches
  - Issue tracking: bug reports, enhancement requests

- **Users @ SNL, LANL, ORNL, NREL, ARL, NRL, U-Utah, ...**

- **Tutorials @ Supercomputing, GPU-Tech, and by arrangement**

- **User Support – DOE ECP; new effort as of Feb’17**
  - On-line user forum
  - Annual bootcamps with tutorials and hackathons
    - ECP applications have priority
  - On-site support for ECP applications at SNL, LANL, and ORNL
Where is Kokkos going? “The Vision”

- **Abstractions** into the ISO/C++ language standard
  - ISO/C++ standard evolving for on-node concurrency & parallelism
  - Voting members include: SNL, ANL, LLNL, LANL, LBNL, ORNL (soon)
  - Advocate for Kokkos abstractions into the standard -> portability
  - Vendors then have ownership of implementation -> performance
  - Standard committee moves slowly...

- **Into Applications written purely in C++11 (or better)**
  - No need for pthreads, OpenMP, OpenACC, OpenCL, CUDA, ...
  - Although these may be used for Kokkos’ back-ends

- **What about Applications written in FORTRAN?**
  - Providing well-defined and proven programming model *abstractions*
  - Perhaps compiler writers could use these for language extensions ...
### What are Kokkos’ Abstractions?

<table>
<thead>
<tr>
<th>Parallel Execution:</th>
<th>Data Structures:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execution Space</strong></td>
<td><strong>Memory Space</strong></td>
</tr>
<tr>
<td>• CPU, MIC, GPU, ...</td>
<td>• Host, HBW, GPU, GPU-shared, ...</td>
</tr>
<tr>
<td><strong>Parallel Patterns</strong></td>
<td><strong>Multidimensional Arrays</strong></td>
</tr>
<tr>
<td>• for, reduce, scan, task-dag</td>
<td>• Traditional (FORTRAN) style</td>
</tr>
<tr>
<td><strong>Execution Policies</strong></td>
<td><strong>Access Policies</strong></td>
</tr>
<tr>
<td>• Range, hierarchical team</td>
<td>• Polymorphic layout</td>
</tr>
<tr>
<td>• Static, dynamic scheduling</td>
<td>• Random, streaming</td>
</tr>
</tbody>
</table>
What is Kokkos’ SCM/SQA?

Software configuration management
Software quality assurance

- **Git on github.com**
- **master** branch: periodic incremental releases
  - User downloads and updates
  - Rigorously tested
- **develop** branch: frequent commits and broad nightly testing
  - Extensive unit tests and mini-app / integration tests
  - Intel Haswell, Intel NKL, ARM v8, IBM Power8, NVIDIA K80, NVIDIA P100, ...
  - GCC (4.8-6.0), Clang (3.6-4.0), Intel (15-17), IBM (13), PGI (17), NVIDIA (7-8), ...
  - Warnings as errors and *extra* warnings
- **develop -> master** promotion process
  - Spotless nightly tests
  - Extensive *upstream* integration testing with Trilinos, LAMMPS, ...
Associated Projects @ Sandia

- **Kokkos Performance Analysis Tools**
  - Runtime hooks for instrumenting parallel dispatch and data management
  - Plug-in at runtime with dynamic link libraries
  - Available: github.com/kokkos/kokkos-tools

- **Kokkos Kernels**
  - Suite of common HPC kernels: linear algebra, graph algorithms, ..
  - Dependence is limited to Kokkos (core)
  - To be available: github.com/kokkos/kokkos-kernels

- **Resource Manager**
  - Broker for reserving on-node execution and memory resources
  - Aware of on-node topology
  - To be available: TBD, talk to us