Kokkos-HClib: enabling high-performance and resiliency for HPC systems

Akihiro Hayashi, Sri Raj Paul, Matthew Whitlock (Georgia Tech)
Nicolas Morales, Jeff Miles, Keita Teranishi (Sandia National Laboratories)
Vivek Sarkar (Georgia Tech)
Kokkos

- What’s Kokkos?
  - Kokkos enables single performance portable codes
    - CPUs, NVIDIA GPUs, ...
  - Kokkos provides data abstractions critical for performance portability not available in OpenMP or OpenACC (Kokkos::View)
  - Essential parallel constructs
    - Kokkos::parallel_for();
    - Kokkos::parallel_reduce();
    - Kokkos::parallel_scan();
Kokkos Example

```cpp
View<int*> x("x", N);  // Construct a View (length: N)

// Parallel For
parallel_for(RangePolicy<>(0, N),
    [=] (int i) {
        x(i) = i;          // C++11 lambda
    });

int result = 0;

// Parallel Reduce
parallel_reduce(RangePolicy<>(0, N),
    [=] (int i, int &update) {
        update += x(i);    // C++11 lambda
    }, result);          // Computational Body

std::cout << result << std::endl;  // Value to update
```
Enabling Resiliency in Kokkos

- **Motivation**
  - Failure probability “significantly” increases as the number of nodes increases [1]

- **Observation**
  - User-level failure mitigation mechanisms will become more important in the next generation systems
    - ✓ Checkpoint/Restart (C/R)
    - ✓ Task Replay
    - ✓ Task Replication
    - ✓ Algorithm-based Fault Tolerance (ABFT)

Enabling Resiliency in Kokkos

Research Question

- When enabling resiliency in Kokkos, what is a desirable point in the performance-portability-productivity (P3) space?
- What is a desirable user-facing parallel-resilience API that enables good productivity-portability?

Original Kokkos Program → Resilient Kokkos Program → Resilient Execution on different platforms

Minimal Modifications

CPUs
- CPU0
- CPU1
- CPU2
- CPU3

GPUs
Resilient Kokkos

- **Overview**
  - Provides Kokkos with resilient execution capability
  - Build on top of Resilient HClib [2]
    - Provides various runtime APIs for task-level resiliency

- **Our Current Focus**
  - Implementation of Kokkos-HClib Backend
  - Design and implementation of Kokkos Task Replay/Replication API

<table>
<thead>
<tr>
<th>parallel_for, parallel_reduce, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenMP Backend</td>
</tr>
<tr>
<td>CUDA Backend</td>
</tr>
<tr>
<td>HPX Backend</td>
</tr>
<tr>
<td>HClib Backend</td>
</tr>
</tbody>
</table>

HClib [2] is a light-weight, work-stealing, resiliency-aware tasking runtime

Building Kokkos-HClib backend

- Non-resilient Execution of miniFE and miniAero of the Mantevo benchmark on Cori@NERSC
- The HClib backend is comparable to the OpenMP backend

CPU: 16-core 2.3 GHz Intel Xeon Processor E5-2698 v3 (Haswell), Compiler: gcc 8.3
# Resilient Kokkos API Design

- **Key Idea:** Provide resilient execution spaces

## Original parallel_for

```cpp
parallel_for(RangePolicy<ExecutionSpace>(0, N), lambda);
```

## Task Replication

- **Semantics:** Replicate a task, and check for the equality of output variables

## Replicate parallel_for

```cpp
parallel_for(RangePolicy<ReplicateSpace>(0, N), lambda);
```

## Task Replay

- **Semantics:** Execute a task, then invoke a user-provided error checking function, and replay the task if the error checking is failed

## Replay parallel_for

```cpp
parallel_for(RangePolicy<ReplaySpace>(0, N), lambda, lambda_check);
```
Preliminary Performance Evaluation (Overhead of Error Checking)

Variants
- Original: parallel_for only
- Replay (bulk): parallel_for w/ a single error checking (no replay)
Summary & Future Work

_summary
- The Kokkos Resilient API enables resiliency in Kokkos in a portable and productive manner
  - Essentially, the user is only supposed to 1) replace an execution policy with a resilient version and 2) provides an error checking function if necessary
- The developed Kokkos-HClib backend is comparable to existing OpenMP backend for non-resilient execution
  - Verified with two of the Mantevo benchmark (miniFE/miniAero)

_future work
- Continue the implementation and discuss the performance impact of the Resilient API using different applications
- Support the all the Kokkos constructs in the Kokkos-HClib backend