Performance Evaluation of the NVIDIA Tesla P100: Our Directive-Based Partitioning and Pipelining vs. NVIDIA’s Unified Memory

Xuewen Cui, Thomas R. W. Scogland, Bronis R. de Supinski and Wu-chun Feng

Abstract
- Heterogeneous supercomputing with accelerators (e.g., GPUs, FPGAs, APUs) continues to increase.
- Programming models for heterogeneous supercomputing (e.g., OpenMP, CUDA, OpenCL) enable offloading of compute-intensive workloads to accelerators.

Motivation
- Drawbacks of directive-based programming models (e.g., OpenMP):
  1. Manual partitioning of data by user whenever device memory exceeded.
  2. Use of the same variable for CPU & GPU in current directive-based extensions limits the potential to split tasks.

Goal
- A new directive-based partitioning and pipelined extension for OpenMP that
  ✓ Automates the overlap of data transfer & kernel computation.
  ✓ Automates the reduction of GPU memory usage.
  ✓ Maps data to a device buffer and automates memory-constrained array indexing and sub-task scheduling.

Summary
- Relative to NVIDIA’s Unified Memory (UM), our directive-based partitioning and pipelined extension on a NVIDIA Pascal P100 system
  ✓ Delivers 68% better performance (on average) for data that fits in GPU memory
  ✓ Delivers 550% better performance (on average) for data that does not fit in GPU memory, particularly for large data sets

Proposed Extension Syntax

```
#pragma omp target
pipeline(schedule_kind(chunk_size,num_stream)) 
pipeline_map(map_type:array_split_list))
pipeline_mem_limit(mem_size)
```

- `pipeline()` inputs
  - `<schedule_kind>`: Scheduler to use for this region (static, adaptive)
  - `<chunk_size>`: Sub-task chunk size
  - `<num_stream>`: Stream number to launch on GPU

- `Pipeline_map()` and `pipeline_mem_limit()` inputs
  - `<map_type>`: to/from/to/from for input/output/input & output arrays
  - `<array_split_list>`: array declaration
  - `<mem_size>`: maximum memory usage
  - `<array_split_list>`: array_split_list structure

- `<<var>`: variable (array) to copy
  - `[split_iter:split_start:offset, size:split_range other non-related dimensions]

Environment Setup and Benchmarks

- CPU: IBM Power8 Processors
- GPU: NVIDIA Tesla P100 16GB with NVLink
- Benchmarks:
  - 3D Convolution
  - Matrix-Multiplication

Performance Results and Conclusions

- Benchmarks:
  - 3D Convolution
  - Matrix-Multiplication

References

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.