Proxy Apps Mysteries Revealed

PI: David Richards (LLNL)
Team: Jeanine Cook (SNL), Hal Finkel (ANL), Christoph Junghans (LANL), Peter McCorquodale (LBNL)
Shirley Moore (ORNL), Omar Aaziz (SNL), Tanner Juedeman (SNL), Courtenay Vaughan (SNL),
Brian Homerding (ANL), Thomas Uram (ANL), Abhinav Bhatel (LLNL), Xavier Andrade (LLNL),
Robert Pavel (LANL), Vinay Ramakrishnaiah (LANL), Tiffany Mintz (ORNL), Greg Watson (ORNL)

Mission

To curate a suite of proxy applications that are representative of the intended characteristics of their respective parent applications and are easy to obtain and use. Characteristics include hardware bottlenecks (e.g., memory, computation, communication) and programming models.

Proxy Catalog Performance Characterization

Latency

- AMG
- CANDLE Benchmarks
- Ember
- ExaMiniMD
- Laghos
- MACSio
- miniAMR
- miniIQMC

Bandwidth

- miniVite
- NEKbone
- PICSARlite
- SW4lite
- SWFFT
- thornado-mini
- XSbench

Compute

Quantitative Assessment

Goal: Understand how well proxies represent parent applications quantitatively at hardware level → better proxies in future

- Representative problems/sizes
- Detailed profiling
- Quantitative characterization
- Statistical proxy/parent app comparison

Quantitative Comparison

Methodology

- Explore how communication of a proxy application relates to its parent application
- Used two quantification methods:
  - A pairwise communication quantification method that captures how much one application matches the other
  - A message characteristics quantification method that produces a clustering-based relatedness measure of the parent applications and their proxies

Calculations

- Contribution percentage
- Prepares and sequences information

Hierarchical clustering

- NET: parent
- SWFFT: proxy for HACC
- SW4: parent
- SW4lite: proxy for SWFFT

GPU Characterization of Proxy Apps

Observations:

- All four proxy applications are limited by performance on P100 and V100 GPUs.
- Applications that are latency-bound on P100 remain latency-bound to same degree on V100, in spite of improvements to cache/memory hierarchy.
- Reduced code size by 80%
- Reduced code size by 80%
- Reduced code size by 80%

Proper Rost Communication Analysis

Proxy App Team worked with ECP Projects to develop or enhance new proxy apps

- PICSARlite: Reduced code size by 80%
- miniVite: Consulted on design of proxy and selection of algorithm
- thornado-mini: Helped with documentation and testing

53 proxy applications now in our catalog. Always looking for more

- Each proxy app can be installed with a simple Spack command
- Dependencies are easily identified and provided

See https://proxyapps.exascaleproject.org/downloads

- Improved metadata to track provenance and sponsors of proxies
- The Proxy App Team worked with ECP Projects to develop or enhance new proxy apps

- PICSARlite: Reduced code size by 80%
- miniVite: Consulted on design of proxy and selection of algorithm
- thornado-mini: Helped with documentation and testing

Added additional information to the proxy app catalog

- Performance bounds and data
- Problem size and run information
- Important code segments with cache and memory bandwidth data