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 Bhatele (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 App Suite v2.0

Quantitative Assessment

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



AMG
CANDLE Benchmarks
Ember
ExaMiniMD
Laghos
MACSio
miniAMR
miniQMC

miniVite NEKbone PICSARlite SW4lite SWFFT thornado-mini XSBench

- 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

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

Quantitative Comparison

• Explored how the communication of a proxy application relates to its parent

Methodology



Proxy Catalog Performance Characterization

Latency

HPCCG

A simple conjugate gradient benchmark code for a 3D chimney domain on an arbitrary number of processo

Problem Size Discussion

From the application README

Suggested: Data size is over a range from 25% of total system memory up to 75%. With exercy-eng and e e ex + exy + eg

Total memory per MPI rank: 728 + n bytes for 27 pt stencil. 248 + n bytes for 7 pt stencil Additional details in application READMI

Analysi

On the Skylake machine on which it was analyzed, HPCCG uses 70-80% of the DRAM bandwidth despits ing memory latency issues due to a indirect memory access.

Parameters

Compiler = icc (ICC) 18.0.1 20171018 Build_Flags = -g -03 -march=native tree-vectorize -qopenmp -DUSING_OMP Run_Parameters = 256 256 256





HPC sparsemv.cpp

const int nrow = (const int) A->local_nrow

86 for (int j=0; j< cur_nnz; j++)</pre>

y[i] = sum;

90 return(0);

sum += cur_vals[j]+x[cur_inds[j]];

SAND No.

Threads (Time)	IPC per Core	Loads per Cycle	L1 Hits per Cycle	L1 Miss Ratio	L2 Miss Ratio	L3 Miss Ratio	L2 B/W Utilized	L3 B/W Utilized	DRAM B/W Utilized
1 (83.1%)	1.39	0.77	0.68	1.88%	42.24%	72.59%	11.36%	38.16%	42.23%
56 (70.4%)	0.47	0.25	0.22	1.74%	42.51%	68.42%	3.27%	28.55%	78.07%
112 (65.5%)	0.48	0.13	0.12	2.19%	42.56%	75.15%	3.51%	28.05%	76.01%

Bandwidth

CloverLeaf

From http://uk-mac.github.io/CloverLea

CloverLeaf is a mini-app that solves the compressible Euler equations on a Cartesian grid, using an explic second-order accurate method. Each cell stores three values: energy, density, and pressure. A velocity vector is stored at each cell corner. This arrangement of data, with some guantities at cell centers, and others at cell corne is known as a staggered grid

complexity. Each kernel loops over the entire grid and updates one (or some) mesh variables, based on a kernel ependent computational stencil. Control logic within each kernel is kent to a minimum, allowing maximum ptimisation by the compiler. Memory is sacrificed in order to increase peformance, and any updates to variable that would introduce dependencies between loop iterations are written into copies of the mesh

Problem Size Discussion

From README.md

There are four standard input files that are recommended for testing 1. clover be short, in - This is not a very sensitive test and the kinetic energy at the end of this run should

be 0.1193E+01. 2. clover bm. in - This runs for 2955 timesteps and is more sensitive than the first test. Through this simulation the whole computational mesh in traversed by a shock and so it is a good test of the parallel mplementation because all internal boundaries will be crossed during the course of the simulation. The fina kinetic energy should be 0.2590E+01 3. clover_bm16_short.in - This is the "socket" test and has a much larger mesh size footprint. The final kinetic energy should be 0.3075E+00 4. clover be16, in This is a fairly long, large mesh run and the kinetic energy at the final time should be 0.4854E+01

Analysis

CloverLeaf becomes bound by the DRAM bandwidth, partly resulting from high cache miss rates on the Skylak machine which this analysis was performed.



Hit Locations

Compute

RSBench

From README.txt

in libm

Carlo neutron transport

Problem Size Discussion

From README.txt Problem size should be scaled using the -o cparticles> parameter. The default is set to 300.000, but full pplication runs may use up to several billion particles per generation

Analysis

RSBench was compute bound on the Skylake machine that this analysis was run on and spent 27.1% of its time

Double Precision	Scalar	1288 Packed	2568 Packed	5128 Packed	Total FLOPS	GFLOPS/sec
PMU	5.090e+11	1.970e+12	1.780e+11	0.000e+00	5.161e+12	2.157e+02
SDE	2.808e+11	1.004e+12	8.095e+10	0.000e+00	2.621e+12	1.095e+02

Intel Software Development Emulator

Intel SDE	R5Bench
Arithmetric Intensity	0.175
FLOPS per Inst	0.83
FLOPS per FP Inst	1.91
Bytes per Load Inst	12.1
Bytes per Store inst	13.0

Roofline – Intel(R) Xeon(R) Platinum 8180M CPU 112 Threads - 56 - Cores 3200.0 Mhz







📕 others 🔳 load store 📒 control flow 📕 integer 📒 floating point



Instruction mix for PENNANT (sedovbig) and Quicksilver (Coral2)



📕 others 🔳 load store 📒 control flow 📕 integer 📒 floating point

Proxy in Parent

100

#msg

100

Parent in Proxy

41.1

#msg

68.7

Parent/Proxy

HACC/

SWFFT

Full Set

SCorr

0.99

PCorr

0.97

Point-to-Point Communication Analysis

Parent in Proxy

PCorr | SCorr

0.84

0.92

Proxy in Parent

PCorr SCorr

0.99



Instruction mix for SW4lite (LOH1-h100)



memory dependence

41.42 %

Stall cycle breakdown for PENNANT gpumain2 kernel on V100

Communication Similarity, Clustering Message Size, and Frequency



~



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

Instruction mix for NEKbone

Observations:

0

100%

30

0

 \square N

က

N

0

- All four proxy apps are limited by memory 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.
 - SW4lite improved from 14% to 16% of peak
 - PENNANT remained at 8%
- Above instruction mixes are shown for P100. Integer instruction counts, which include address calculations, are reduced and can be performed simultaneously with floating point operations on V100 and thus have less impact on V100.



NNS

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.











instruction fetch 20.52 %

memory dependency 41.42 %