SAND2020-1245PE

EXASCALE COMPUTING PROJECT

Using Cosine Similarity to Quantify Representativeness of ECP Proxy Apps

Jeanine Cook (SNL) and Jefferey Kuehn (LANL)
Omar Aaziz (SNL)
Courtenay Vaughan (SNL)

Motivation for Examining Representativeness

- Proxy applications used for
  - Long term vendor collaboration projects (e.g., PathForward)
  - Procurements (benchmarking/performance estimation)
  - Testing new systems/architectures
- Incentive to limit the number of proxy codes
  - Constrained on staff and time (labs & vendors)
  - Vendors have limited time & staff to respond to RFPs
- Qualitatively down-select number of project codes
  - Debate among team of SMEs about perceived relevance
  - Choices often advocated based on familiarity, ease, etc

Strategy: Add quantitative support to balance qualitative inputs

Insights

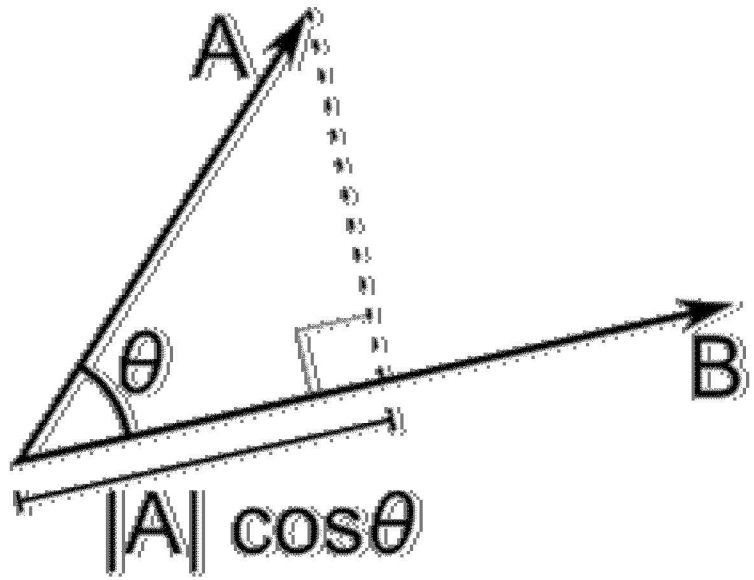
- Performance is interaction of workload with set of design constraints imposed by a particular system
  - Manner and proportion that design constraints affect particular workload becomes the workload fingerprint
- Similar workload fingerprints mean workload responds similarly to particular design constraint and to changes in that particular constraint
  - E.g. Expect codes with similar dependence/bottleneck on memory bandwidth to derive similar benefit from memory bandwidth improvement
- Workload fingerprints must be easy and fast to collect
  - Not through detailed simulators!

Approach

- Rely on two-elements as building-blocks/tools
  - Ability to collect fingerprint for a code
  - Ability to quantify similarity comparison of two fingerprints
- Fingerprint construction
  - Aggregation of set of metrics relevant to system design constraints
    - Hardware performance counters/events grouped by design constraints
      - E.g., Processor frontend, execution, backend, cache/memory hierarchy
- Cosine similarity comparison
  - Compares vectors of performance counter events in high dimensional space

Cosine Similarity

- Is a property of dot (inner) product in vector spaces in two or more dimensions
  - Think: “Projection of **A** in the direction of **B**”
- Uses $\cos\theta$ as an angular distance metric
- Quantifies distance of A and B independent of their magnitude



$$\boldsymbol{A} \cdot \boldsymbol{B} \equiv \sum_{i=1}^{n} a_i b_i = \| \boldsymbol{A} \| \| \boldsymbol{B} \| \cos\theta$$

$$\therefore \cos\theta = \frac{\left(\sum_{i=1}^{n} a_i b_i\right)}{\left(\| \boldsymbol{A} \| \| \boldsymbol{B} \|\right)}$$

Performance Counter Events & Selectivity

Cache	Selectivity
MEM_LOAD_UOPS_L3_HIT_RETIRED.XSNP_HIT	2.721
MEM_LOAD_UOPS_L3_HIT_RETIRED.XSNP_HITM	2.213
MEM_LOAD_UOPS_L3_HIT_RETIRED.XSNP_MISS	2.178
L2_LINES_IN.I	1.531
MEM_LOAD_UOPS_RETIRED.L3_MISS	1.482
L2_RQSTS.RFO_HIT	1.410
L2_RQSTS.CODE_RD_MISS	1.406
MEM_LOAD_UOPS_RETIRED.L2_MISS	1.383
MEM_LOAD_UOPS_L3_HIT_RETIRED.XSNP_NONE	1.305
MEM_LOAD_UOPS_RETIRED.L3_HIT	1.305
L2_LINES_IN.S	1.267
ICACHE.MISSES	1.131
L2_RQSTS.ALL_CODE_RD	1.073
L2_TRANS.CODE_RD	1.070
MEM_LOAD_UOPS_L3_MISS_RETIRED.LOCAL_DRAM	1.067
ICACHE.HIT	1.023
L2_RQSTS.DEMAND_DATA_RD_HIT	1.018
L2_RQSTS.DEMAND_DATA_RD_MISS	0.999

Pipeline	Selectivity
FP_ASSIST.ANY	3.162
FP_ASSIST.X87_INPUT	3.162
MEM_UOPS_RETIRED.STLB_MISS_LOADS	2.839
MEM_UOPS_RETIRED.STLB_MISS_STORES	2.577
LD_BLOCKS.STORE_FORWARD	2.212
UOPS_ISSUED.SINGLE_MUL	2.114
LD_BLOCKS.NO_SR	2.039
UOPS_ISSUED.FLAGS_MERGE	1.977
ILD_STALL.LCP	1.796
DSB2MITE_SWITCHES.PENALTY_CYCLES	1.777
DSB2MITE_SWITCHES	1.777
MISALIGN_MEM_REF.STORES	1.656
LSD.CYCLES_4_UOPS	1.650
LSD.UOPS	1.608
LSD.ACTIVE	1.580
ARITH.FPU_DIV_ACTIVE	1.551
UOPS_DISPATCHES_CANCELLED.SIMD_PRF	1.434
BACLEARS.ANY	1.358

BROADWELL

	ExaMiniMD	LAMMPS	MiniQMC	QMCPack	sw4lite	sw4	SWFFT	HACC	pennant	snap
ExaMiniMD	0.00	10.24	84.61	83.55	61.94	64.17	86.71	85.58	75.88	44.50
LAMMPS	10.24	0.00	75.12	73.95	53.63	56.50	79.66	78.51	70.97	34.97
MiniQMC	84.61	75.12	0.00	5.97	42.91	47.75	51.57	51.28	66.16	43.41
QMCPack	83.55	73.95	5.97	0.00	37.71	42.28	45.85	45.52	60.31	40.89
sw4lite	61.94	53.63	42.91	37.71	0.00	6.47	27.99	26.86	30.17	24.55
sw4	64.17	56.50	47.75	42.28	6.47	0.00	23.59	22.42	23.83	29.89
SWFFT	86.71	79.66	51.57	45.85	27.99	23.59	0.00	1.22	18.65	51.79
HACC	85.58	78.51	51.28	45.52	26.86	22.42	1.22	0.00	18.14	50.70
pennant	75.88	70.97	66.16	60.31	30.17	23.83	18.65	18.14	0.00	51.63
snap	44.50	34.97	43.41	40.89	24.55	29.89	51.79	50.70	51.63	0.00

SKYLAKE

	ExaMiniMD	LAMMPS	MiniQMC	QMCPack	sw4lite	sw4	SWFFT	HACC	pennant	snap
ExaMiniMD	0.00	8.97	81.96	68.83	38.66	39.55	28.51	37.76	43.58	22.20
LAMMPS	8.97	0.00	81.38	68.47	38.60	39.33	29.50	38.49	42.40	20.45
MiniQMC	81.96	81.38	0.00	16.35	47.28	47.63	58.78	49.85	46.58	65.55
QMCPack	68.83	68.47	16.35	0.00	36.05	36.40	46.19	37.82	36.33	53.30
sw4lite	38.66	38.60	47.28	36.05	0.00	4.05	20.56	17.09	12.89	21.69
sw4	39.55	39.33	47.63	36.40	4.05	0.00	19.82	15.87	11.91	22.79
SWFFT	28.51	29.50	58.78	46.19	20.56	19.82	0.00	10.33	24.49	21.44
HACC	37.76	38.49	49.85	37.82	17.09	15.87	10.33	0.00	19.92	26.67
pennant	43.58	42.40	46.58	36.33	12.89	11.91	24.49	19.92	0.00	25.00
snap	22.20	20.45	65.55	53.30	21.69	22.79	21.44	26.67	25.00	0.00

Gaps & Redundancy

	SWFFT	HACC	pennant	sw4	sw4lite	snap	QMCPack	MiniQMC	LAMMPS	ExaMiniMD
SWFFT	0.00	1.22	18.65	23.59	27.99	51.79	45.85	51.57	79.66	86.71
HACC	1.22	0.00	18.14	22.43	26.86	50.70	45.52	51.28	78.51	85.58
pennant	18.65	18.14	0.00	23.83	30.17	51.63	60.31	66.16	70.97	75.88
sw4	23.59	22.43	23.83	0.00	6.47	29.89	42.28	47.75	56.50	64.17
sw4lite	27.99	26.86	30.17	6.47	0.00	24.55	37.71	42.91	53.63	61.94
snap	51.79	50.70	51.63	29.89	24.55	0.00	40.89	43.41	34.97	44.50
QMCPack	45.85	45.52	60.31	42.28	37.71	40.89	0.00	5.97	73.95	83.55
MiniQMC	51.57	51.28	66.16	47.75	42.91	43.41	5.97	0.00	75.12	84.61
LAMMPS	79.66	78.51	70.97	56.51	53.63	34.97	73.95	75.12	0.00	10.24
ExaMiniMD	86.71	85.58	75.88	64.17	61.94	44.50	83.55	84.61	10.24	0.00

Performance Group Breakdown: Cache

	ExaMiniMD	LAMMPS	MiniQMC	QMCPack	sw4lite	sw4	SWFFT	HACC	pennant	snap
ExaMiniMD	0.00	5.02	54.54	38.73	11.70	12.49	6.58	6.38	13.21	7.13
LAMMPS	5.02	0.00	54.69	38.62	15.66	16.27	4.87	6.38	13.60	10.88
MiniQMC	54.54	54.69	0.00	17.15	47.12	46.08	50.02	48.98	42.16	49.15
QMCPack	38.73	38.62	17.15	0.00	32.64	31.67	33.92	32.94	26.29	33.78
sw4lite	11.70	15.66	47.12	32.64	0.00	1.15	13.41	11.40	11.15	5.07
sw4	12.49	16.27	46.08	31.67	1.15	0.00	13.74	11.70	10.69	5.69
SWFFT	6.58	4.87	50.02	33.92	13.41	13.74	0.00	2.24	9.09	8.80
HACC	6.38	6.38	48.98	32.94	11.40	11.70	2.24	0.00	7.86	6.87
pennant	13.21	13.60	42.16	26.29	11.15	10.69	9.09	7.86	0.00	9.37
snap	7.13	10.88	49.15	33.78	5.07	5.69	8.80	6.87	9.37	0.00

Performance Differences with Different Inputs

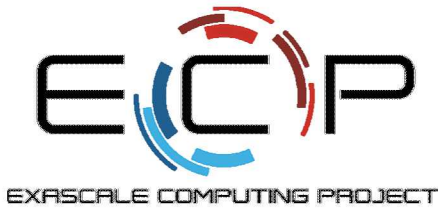
	Angular difference in signatures for clamr_mpiopenmponly -n_4000_-i_100_-t_600					
	regular-grid	regular-grid-by-faces	face-in-place	cell	face	cell-in-place
regular-grid	0.00	0.15	0.19	0.12	0.28	0.27
regular-grid-by-faces	0.15	0.00	0.13	0.16	0.20	0.19
face-in-place	0.19	0.13	0.00	0.19	0.18	0.19
cell	0.12	0.16	0.19	0.00	0.27	0.25
face	0.28	0.20	0.18	0.27	0.00	0.14
cell-in-place	0.27	0.19	0.19	0.25	0.14	0.00
sum	0.99	0.83	0.87	0.98	1.06	1.03
		Best representatives			**Worst representative**	
		Note the six CLAMR methods have **VERY** similar fingerprints				

How Might this be Used?

- Identify **gaps** in representation for set of proxies
- Identify **redundancies** in set of proxies
- Quantify **similarities** between proxies and parents or workloads
  - Infer relationships between proxy and workload performance
  - Infer relationships for particular proxy/parent with varying problem/input
- Apply these three properties to:
  - Provide feedback to proxy developers to improve representativeness
  - Help procurement/project teams to better identify minimum spanning sets
  - Identify workload-platform mappings by similarity
    - Identify workloads that are favorable candidates to port to GPU
    - Steer application workloads toward favorable architectures

Future Work

- Infer error bounds on similarity-based proxy performance projections
- Validation
  - Correlate results with additional performance data
- Examine network and I/O behavior similarity
- Determine which applications optimally map to which architectures based on similarity
- Predict porting effort to target architectures
  - Quantify code differences in application ports to target architectures
  - Use application similarity to predict potential code effort



Thank you!