Porting mini-apps to ARM HPC systems

Brian J Gravelle
Dave Nystrom

September 2019
ARM Research Summit
(unclassified)

Introduction

- x86 and Power dominate HPC CPU market

- ARM is new alternative with potential for
  - Low-power systems
  - Customization through multiple chip makers
  - High levels of parallelism

Introduction

How will old codes work on new systems?

Are the performance issues significantly different?

Systems Used

- Compare Intel Skylake to Marvell Thunder X2

	Skylake Gold 6152	ThunderX2-B1
Cores	44	56
Threads per core	2	4
Clock	2.1GHz	2.0GHz
L1 data cache	32K	32K
L2 cache	1024K	256K
L3 cache	30976K	32768K
Memory Controllers	6	8
SIMD instructions	Up to 512 bit	128 bit NEON

*This work used systems funded by the Computational Systems and Software Environments (CSSE) subprogram of LANL's ASC program NNSA/DOE

Measurement Methodology

- TAU
  - Performance measurement
  - Sampling
  - Profiling
- Caliper
  - Performance measurement
  - Instrumentation
- PAPI
  - Hardware counter interface

Measurements of Interest

- Frontend and Backend Stalls
- Cache performance
- SIMD instruction use
- Energy

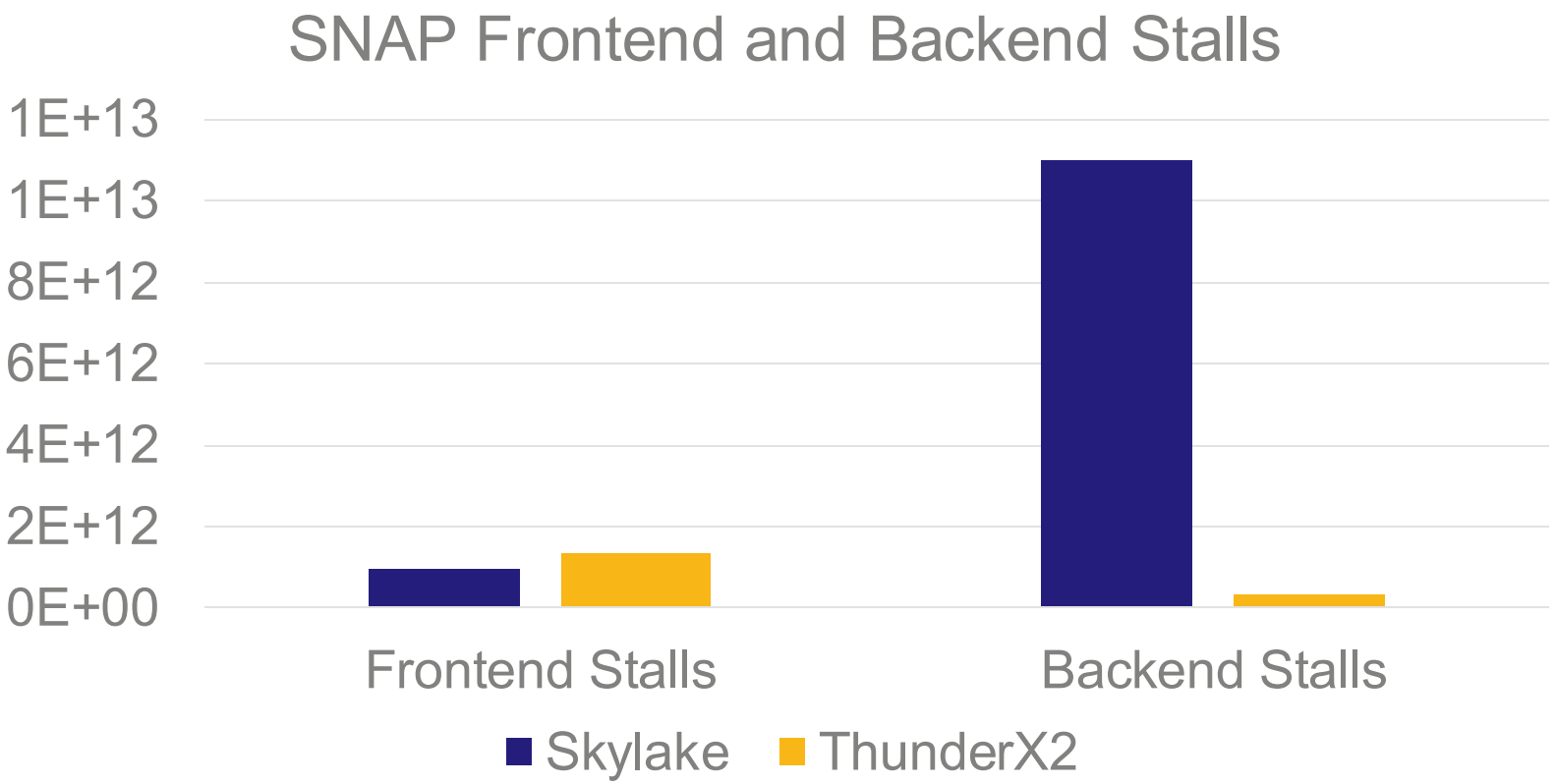
Mini-apps used

- SNAP
  - Mini app for PARTISN
  - Computation based on 6D neutral particle transport
  - 3D structured spatial mesh
  - 3D velocity space uses 2 angle coordinates and an energy coordinate
  - Fortran, MPI, OpenMP

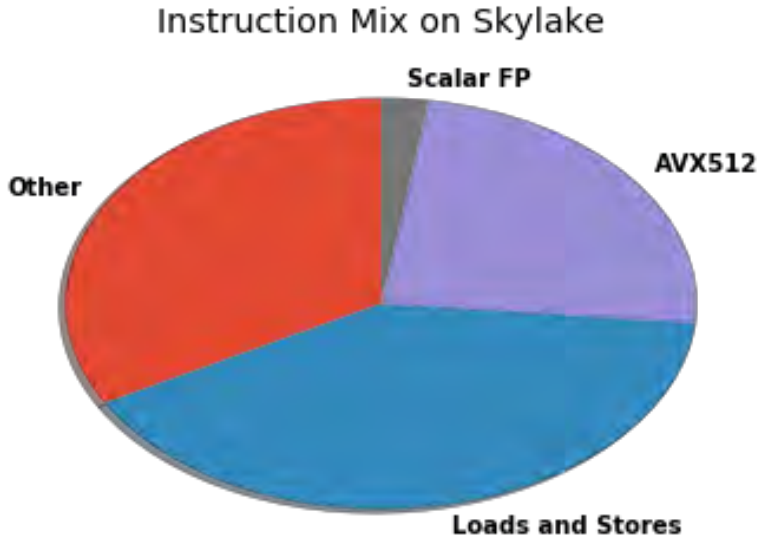
SNAP Problem size

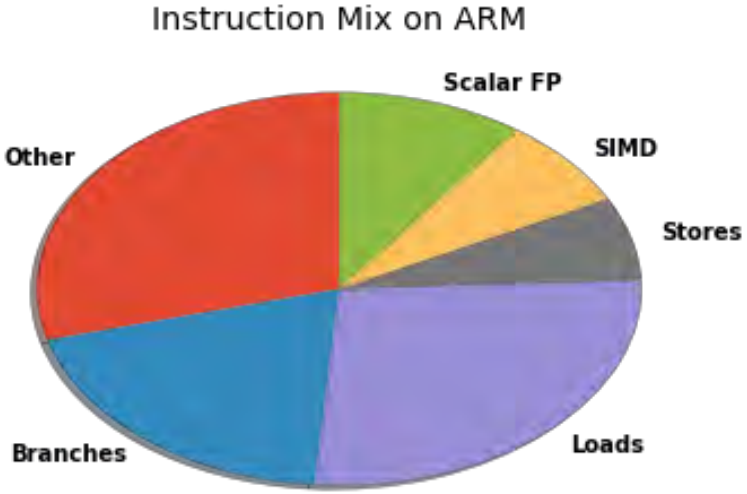
- Problem Size
  - 3D mesh 270x40x64
- Skylake
  - MPI+openmp threading
  - 40 ranks with 2 threads each
- ThunderX2
  - MPI+openmp threading
  - 40 ranks with 4 threads each

SNAP Analysis (Intel 2t ARM 4t)

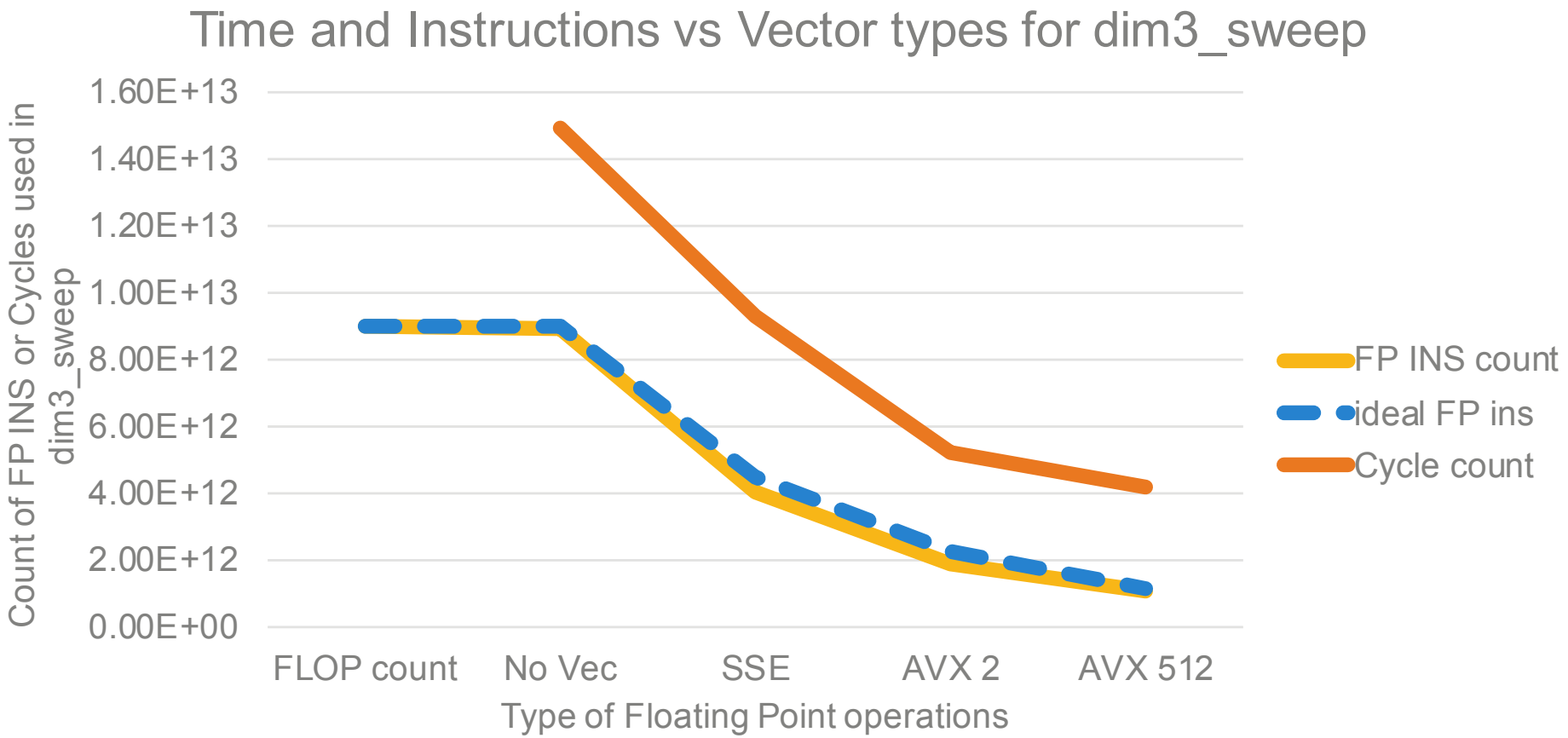


SNAP Analysis (Intel 2t ARM 4t)





SNAP Analysis (Intel 2t)



SNAP Analysis (ARM 4t)

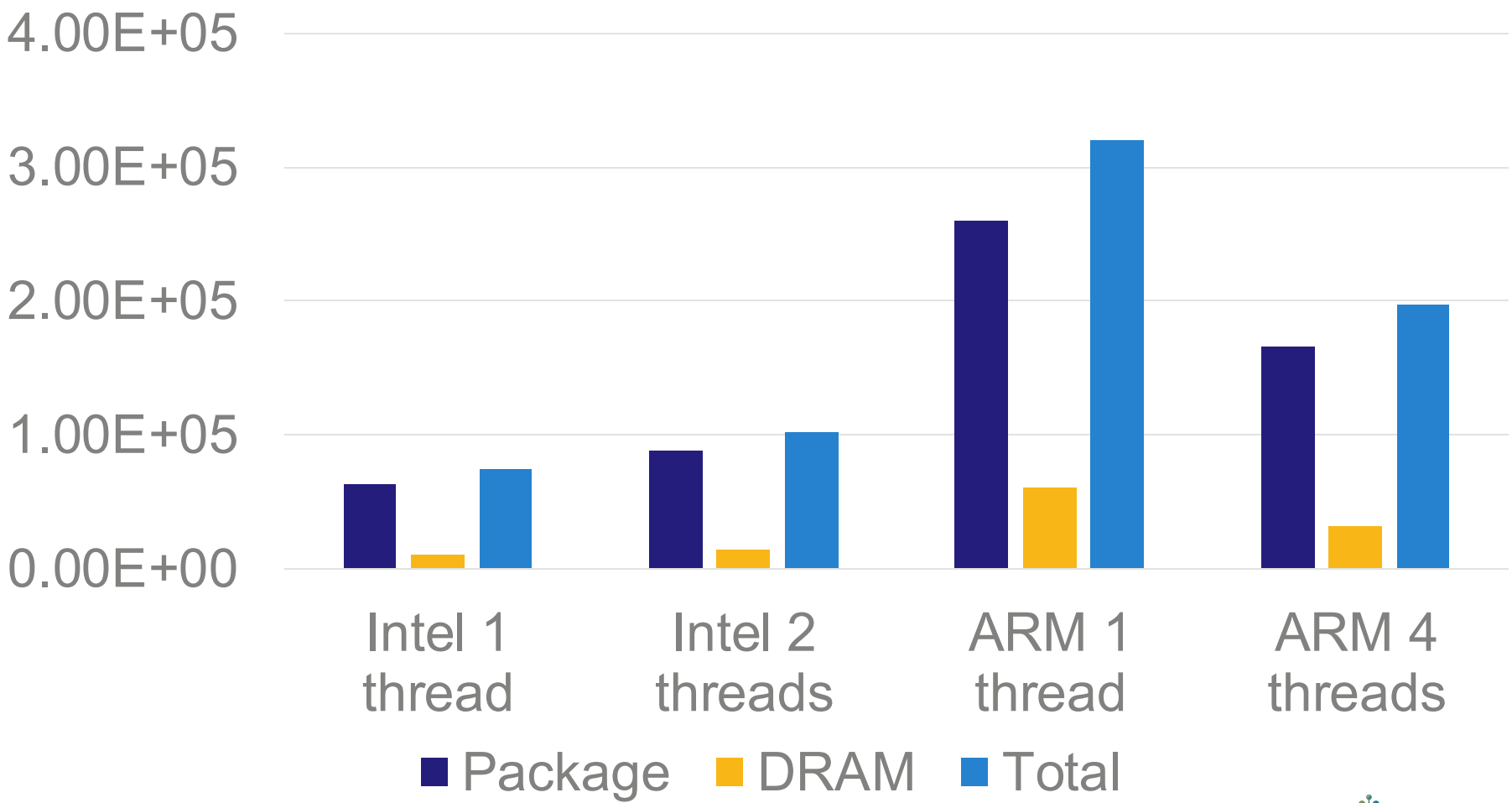
- ARM SIMD comparison – solve time

No SIMD	NEON	Speedup
84.7s	66.1s	1.28x

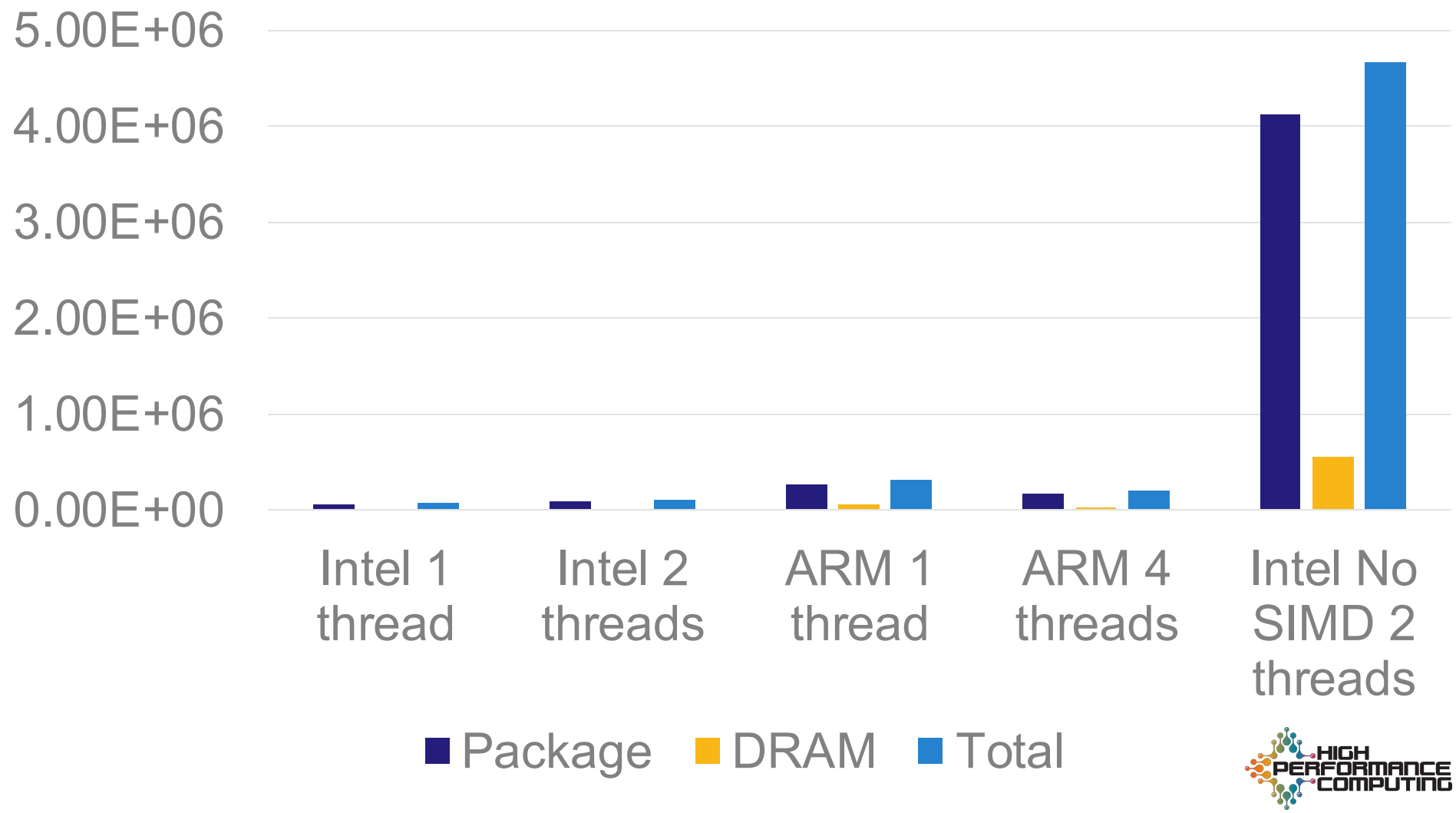
SNAP Analysis

- Vector instructions provide significant improvement to Skylake performance
- Not so much for ThunderX2

SNAP Energy Results



SNAP Energy Results



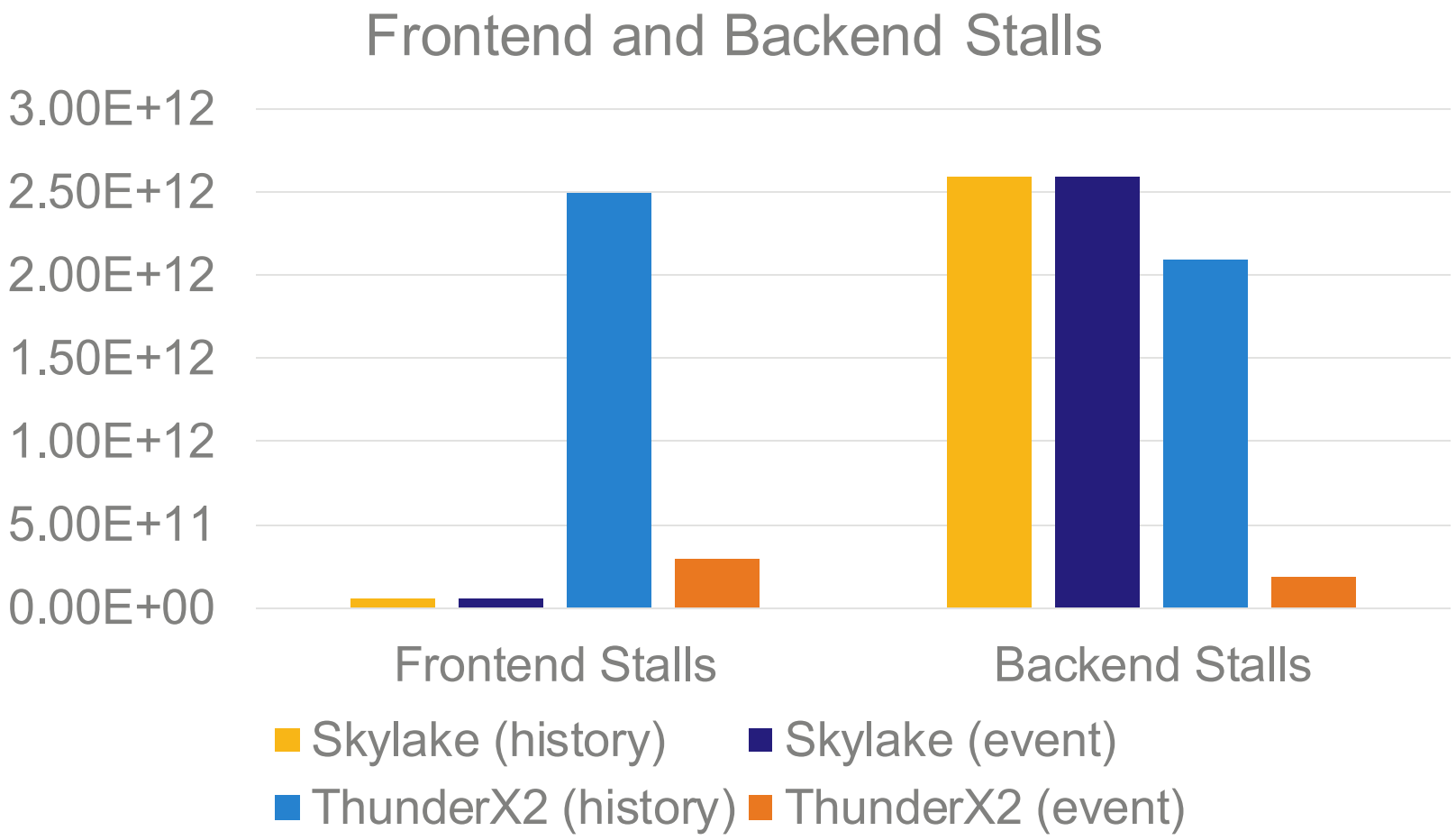
Mini-apps used

- XSBench
- “mini-app representing a key computational kernel of the Monte Carlo neutronics application OpenMC”
- Mostly tabular lookup based on randomly generated energy values
- C, OpenMP

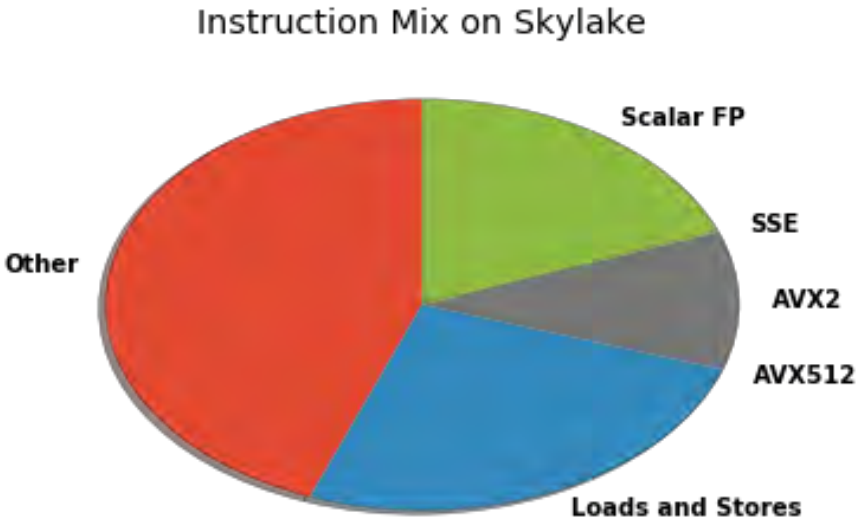
XSBench Problem size

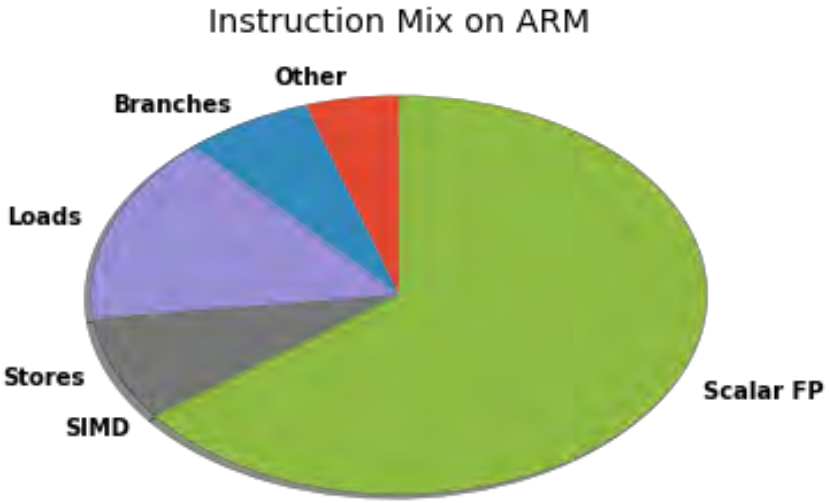
- Problem Size
  - Standard size reactor with 5E6 particles
- Skylake
  - OpenMP threading
  - 88 threads
- ThunderX2
  - OpenMP threading
  - Ranging from 56 to 448 threads

XSBench Analysis

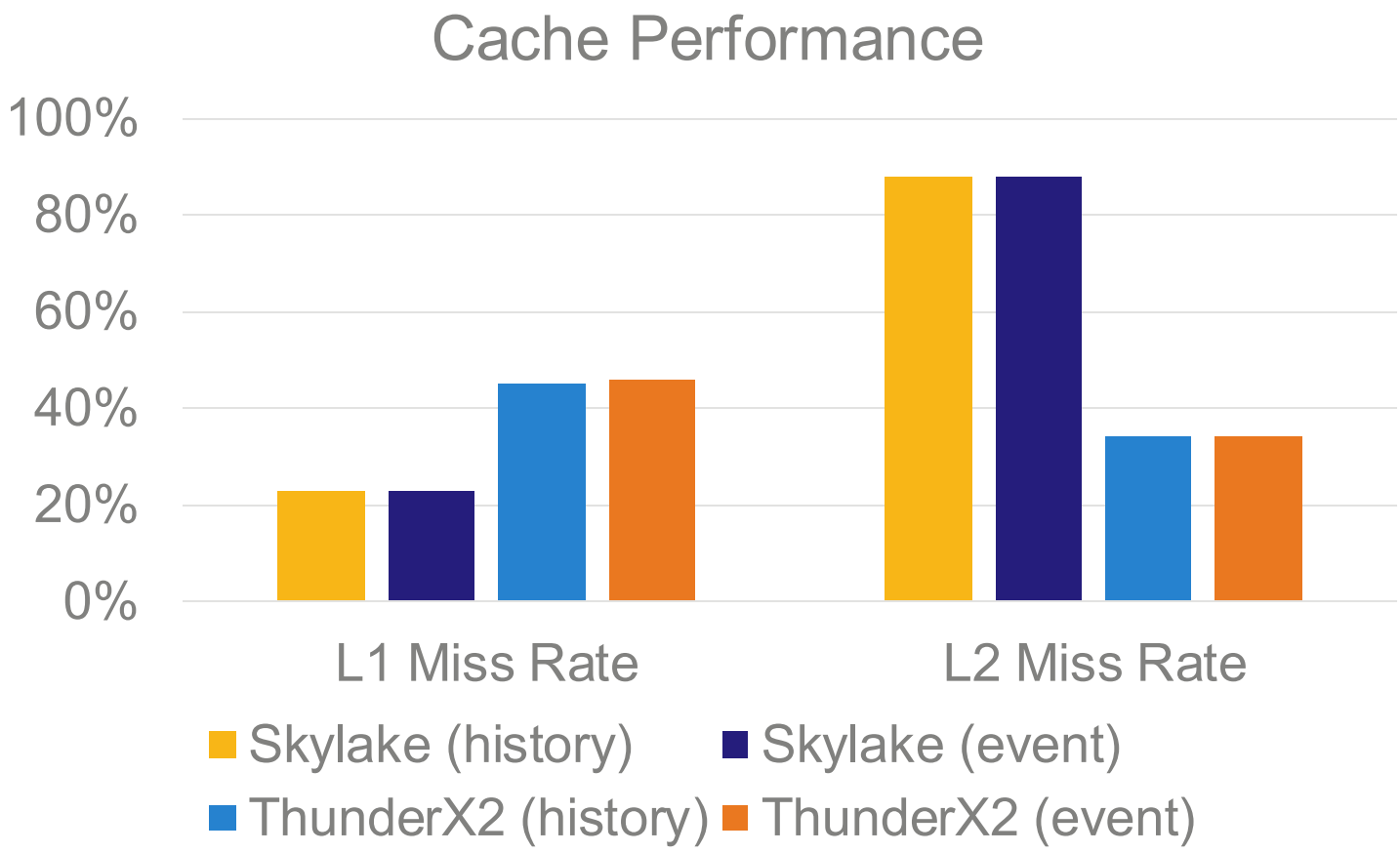


XSBench Analysis





XSBench Analysis



XSBench Improvements

- Event Based XSBench
  - There is a nuclide array with each element storing random energies other values
  - The main loop iterates over this array
  - For each nuclide it looks up data based primarily on the energy value
  - Lookups are distributed to threads
  - Randomly ordered energy values prevent locality in the lookups

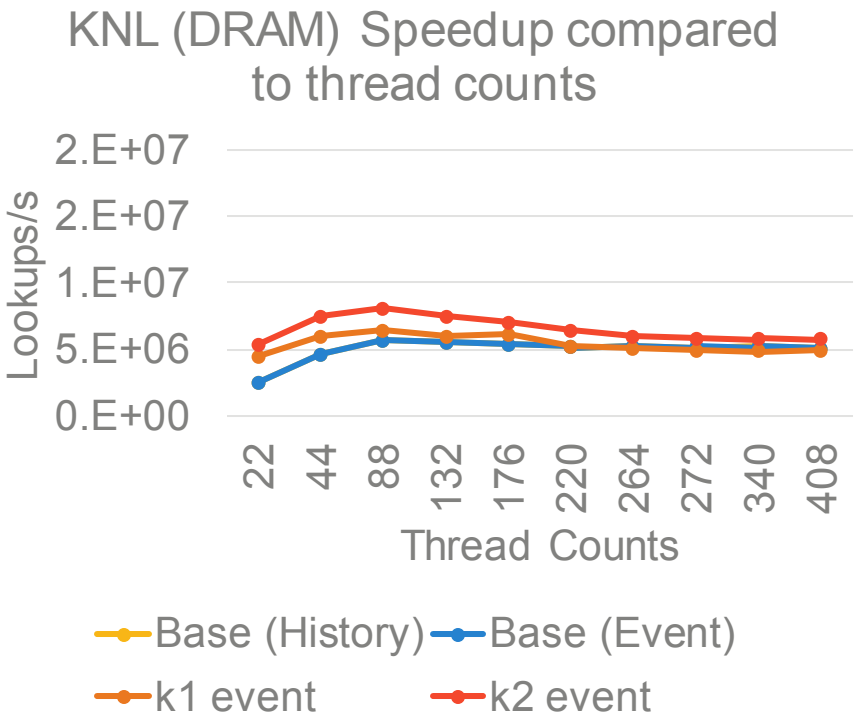
XSBench Improvements

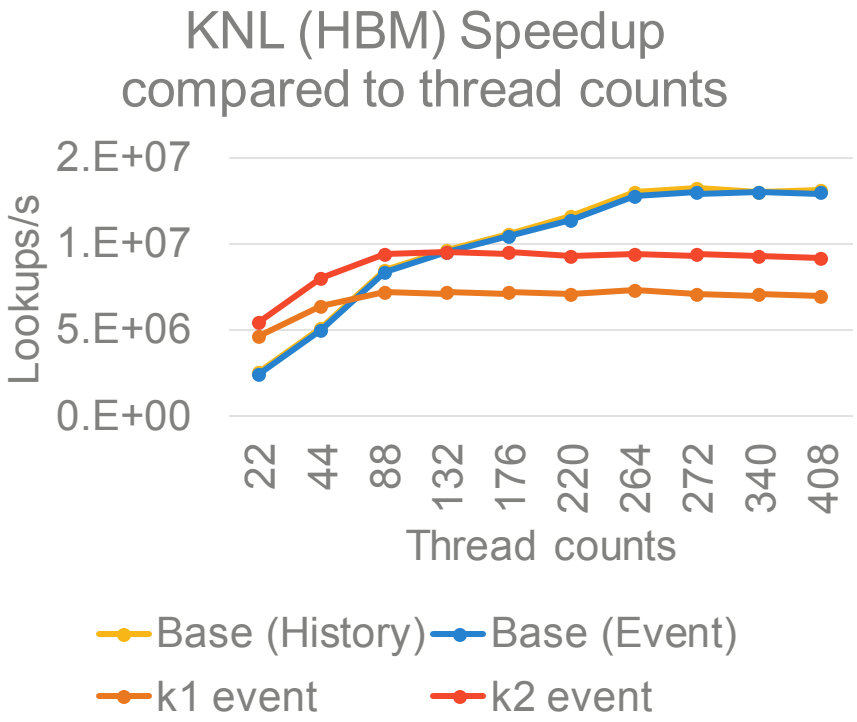
- To improve cache locality sort the nuclide array before performing lookups
- Optimized kernel in distribution sorts based on the energy and the material
- Our version sorts only based on the energy

XSBench Improvements

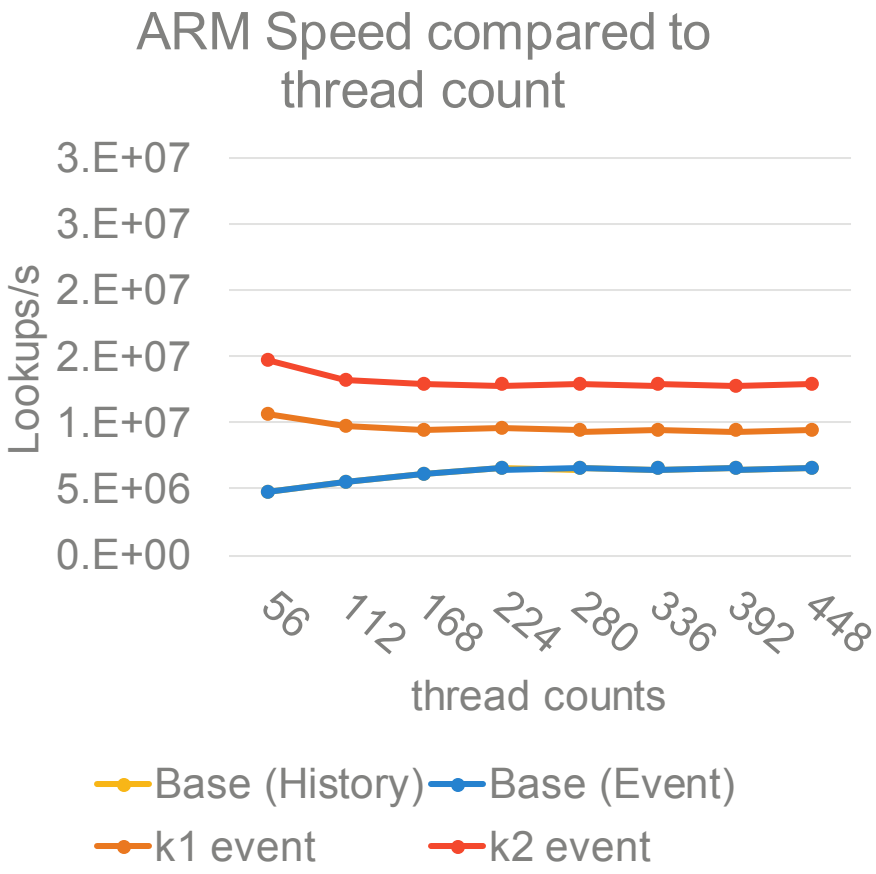
- “Base” versions are the default (event or history)
- “K1” – optimized event-based in distribution
- “K2” – event-based version optimized by us

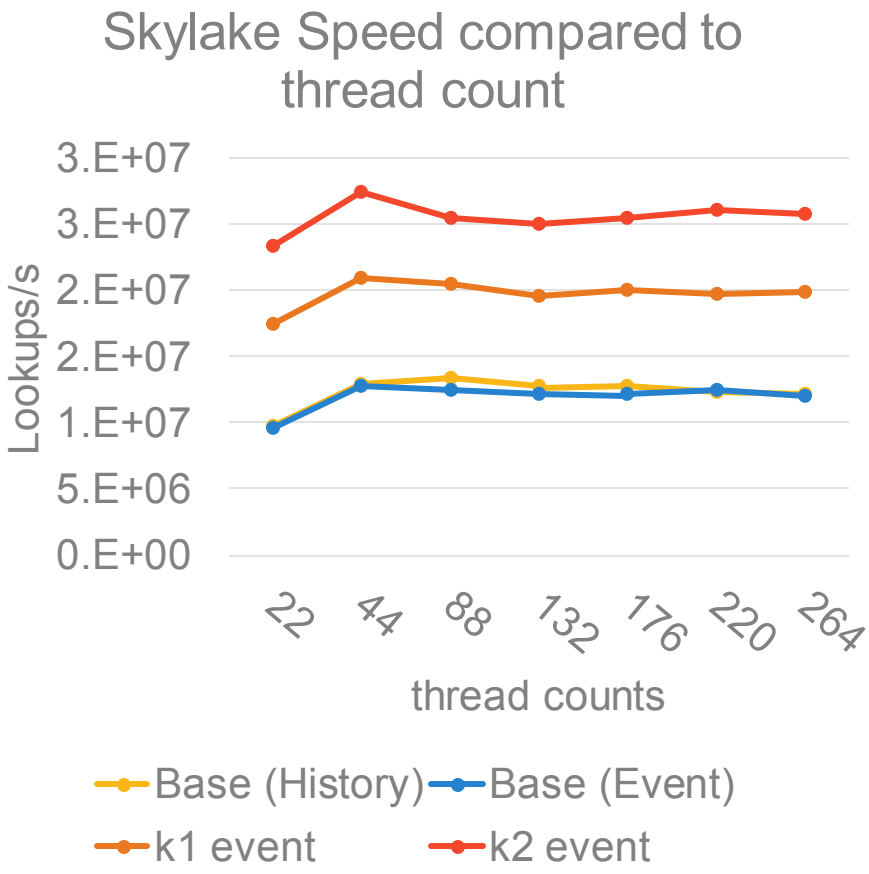
KNL Experiment



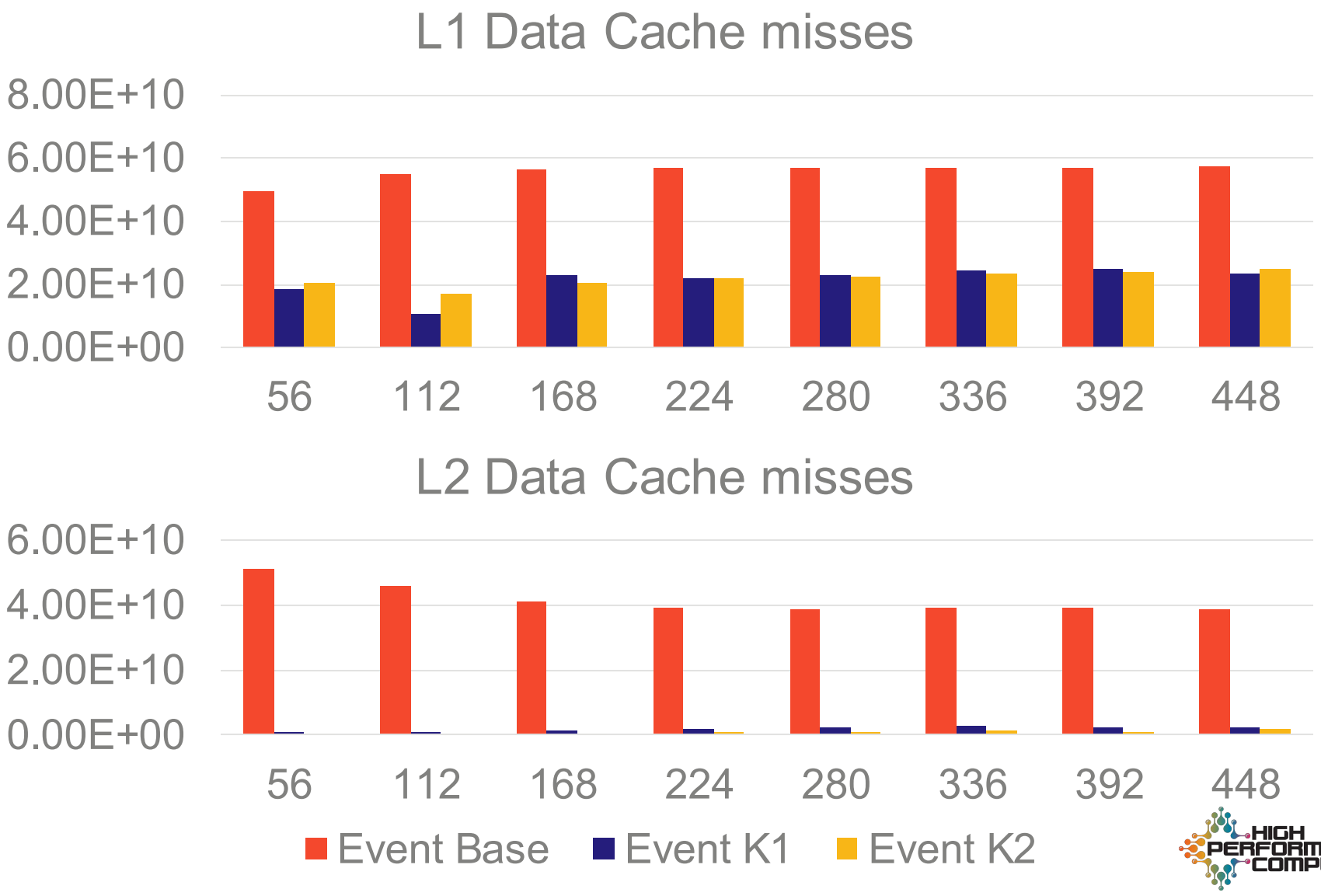


Results

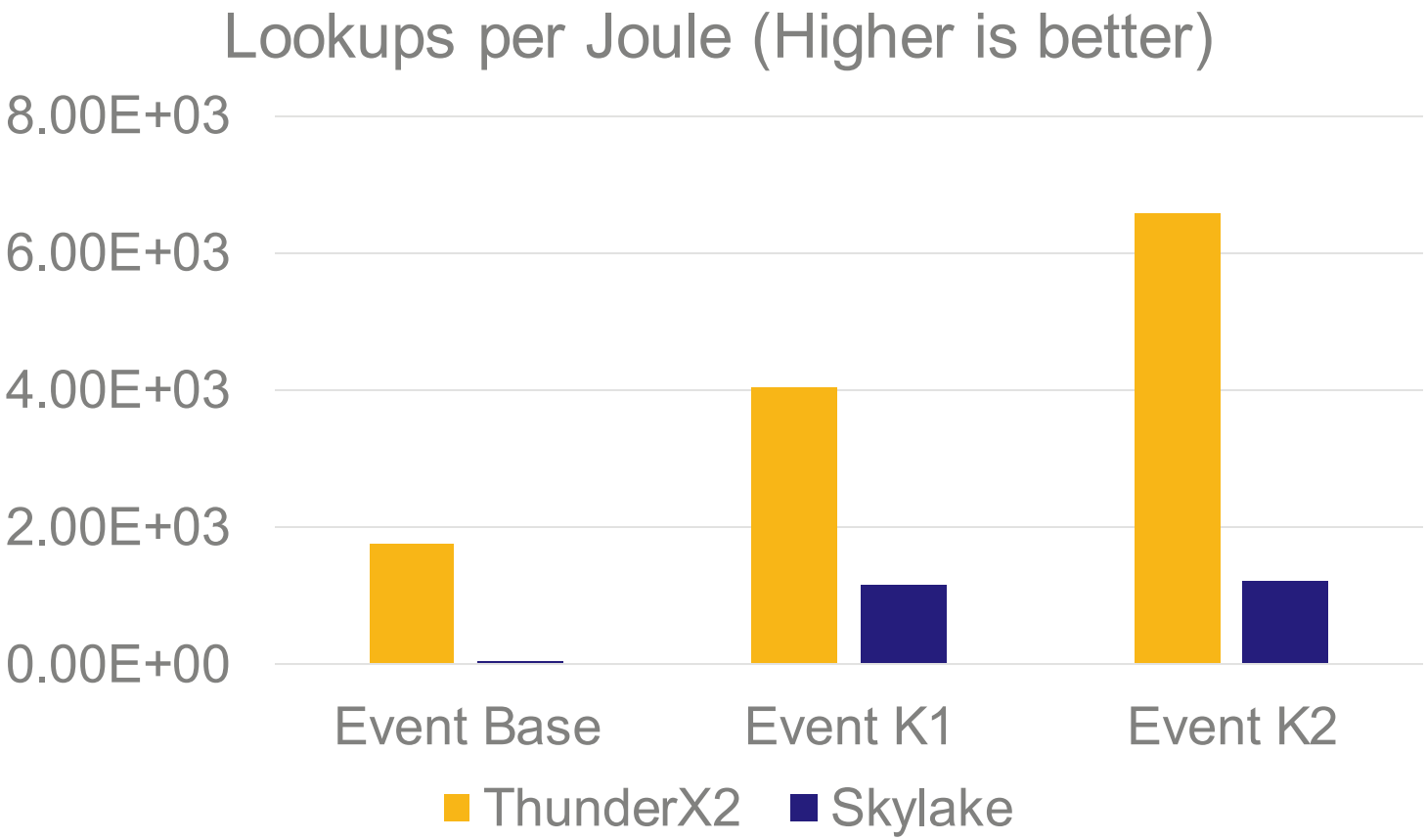

ARM Speed compared to thread count
Lookups/s
3.E+07
3.E+07
2.E+07
2.E+07
1.E+07
5.E+06
0.E+00
56
112
168
224
280
336
392
448
thread counts
Base (History)
Base (Event)
k1 event
k2 event


Skylake Speed compared to thread count
Lookups/s
3.E+07
3.E+07
2.E+07
2.E+07
1.E+07
5.E+06
0.E+00
22
44
88
132
176
220
264
thread counts
Base (History)
Base (Event)
k1 event
k2 event

ARM HW Counter comparison



XSBench Energy Results



Conclusion

- Getting mini-apps (and presumably the full applications on which they are based) up and running on ARM is easy.
- Performance bottlenecks are for the most part similar; i.e. memory bound vs compute bound does not change.
- Some specific issues may differ; in these cases frontend stalls were higher than backend on the ThunderX2.

Questions?

Over 70 years at the forefront of supercomputing