Recent progress on the ACRANEB2* dwarf from the ESCAPE project, part II

Jacob Weismann Poulsen and Per Berg
DMI

*1) Single interval shortwave radiation scheme with parameterized optical saturation and spectral overlaps by J. Masek et al, Q. J. R. Meteorol. Soc. (2015) DOI:10.1002/qj.2653
*2) Single interval longwave radiation scheme based on the net exchanged rate decomposition with bracketing by J.F. Geleyn et al, Q. J. R. Meteorol. Soc. (2017) DOI:10.1002/qj.3006

Acknowledgement:
Bent Hansen Sass and Kristian Pagh Nielsen (DMI)
Peter Messmer and Stan Posey (NVIDIA)
Mike Greenfield and Karthik Raman (Intel)

Refactor your code - do it now or do it later!

Algorithm

Implementation

Silicon

1E+7
1E+6
1E+5
1E+4
1E+3
1E+2
1E+1
1E+0
1E-1

1970 1975 1980 1985 1990 1995 2000 2005 2010 2015

Transistors (thousands)

Single-thread performance (SpecINT)

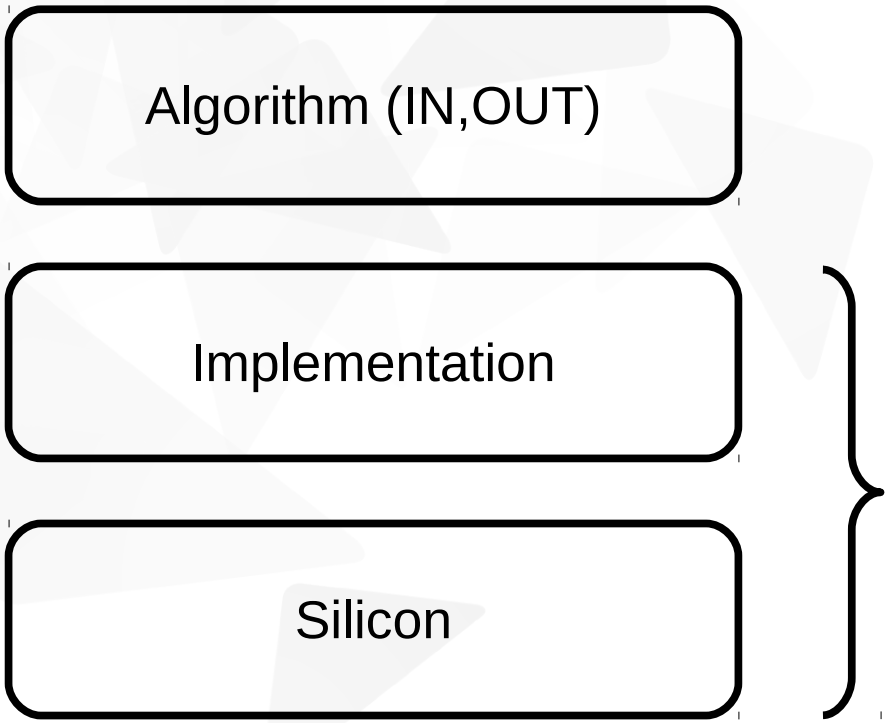
Frequency (MHz)

Typical power (watts)

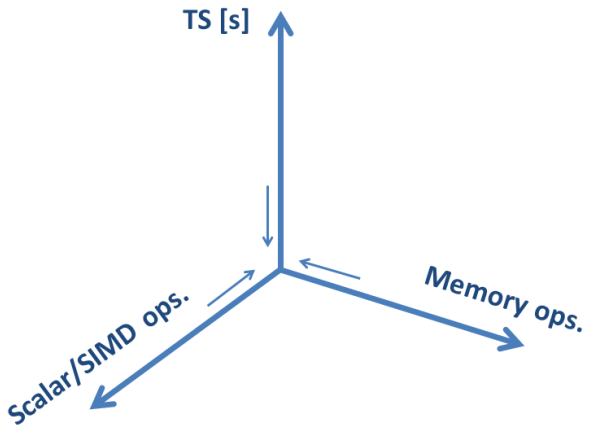
Number of cores

Kirk M. Bresniker, Sharad Singhal, R. Stanley Williams, "Adapting to Thrive in a New Economy of Memory Abundance", Computer, vol. 48, no. , pp. 44-53, Dec. 2015, doi:10.1109/MC.2015.368

Minimal problem: Memory operations



Operation	Energy [pJ]	Time [nsec]
64 bit FMA	200	1
Read 64 bits from cache	800	3
Read 64 bits from DRAM	12000	70



- ESCAPE focus: multi-core processors (proper baseline), many-core processors and many-core accelerators.
- Two-step approach:
- 1) Reduce implementation overhead to establish a baseline
- 2) Expose parallelism at all levels (threads/SIMD/SIMT)

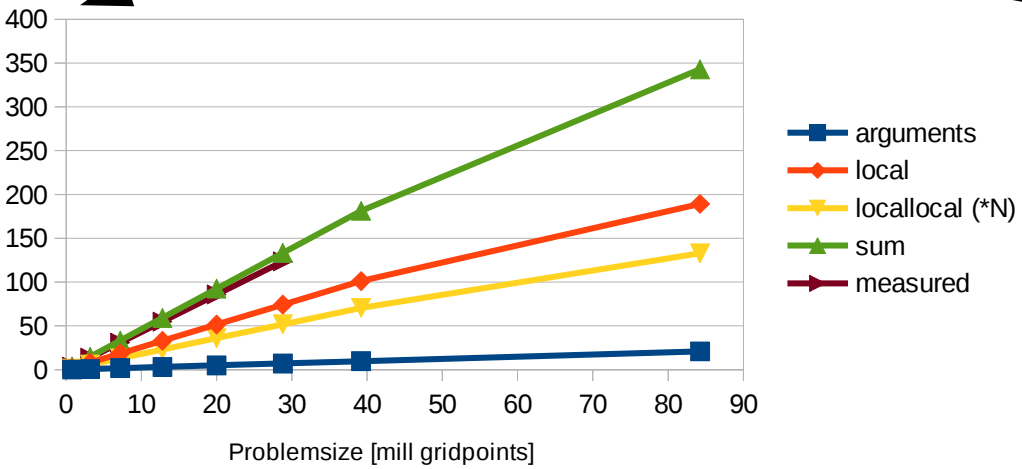
Memory – approach minimal problem

Baseline	case [s]
SLOC [lines]	5687
Language state	F77 fixed-size, cast
Technical state	OK
Numerical state	4 digits
Largest psize on 16Gb	200x200x80
Largest psize on 64Gb	400x400x80
Largest psize on 128Gb	600x600x80
Target psize	1200x1080x65

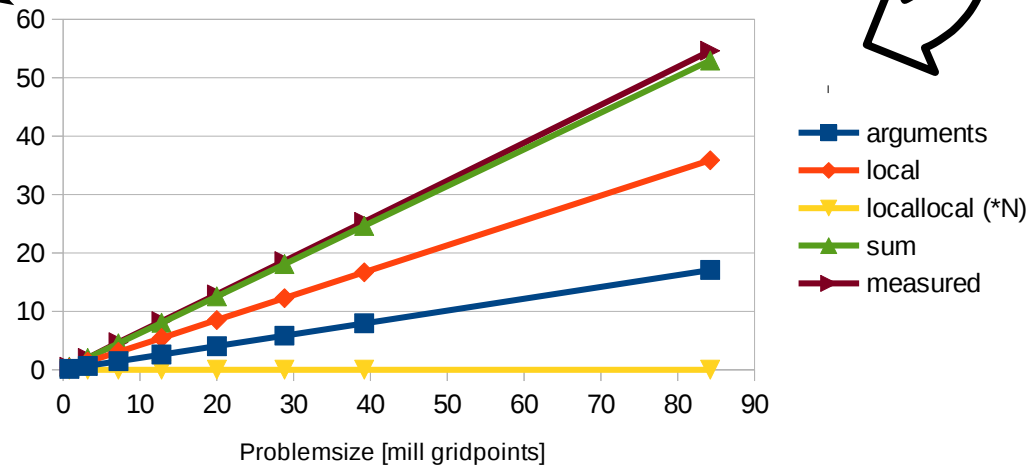
1653

Now, < 64Gb

Memory footprint [Gb] in baseline

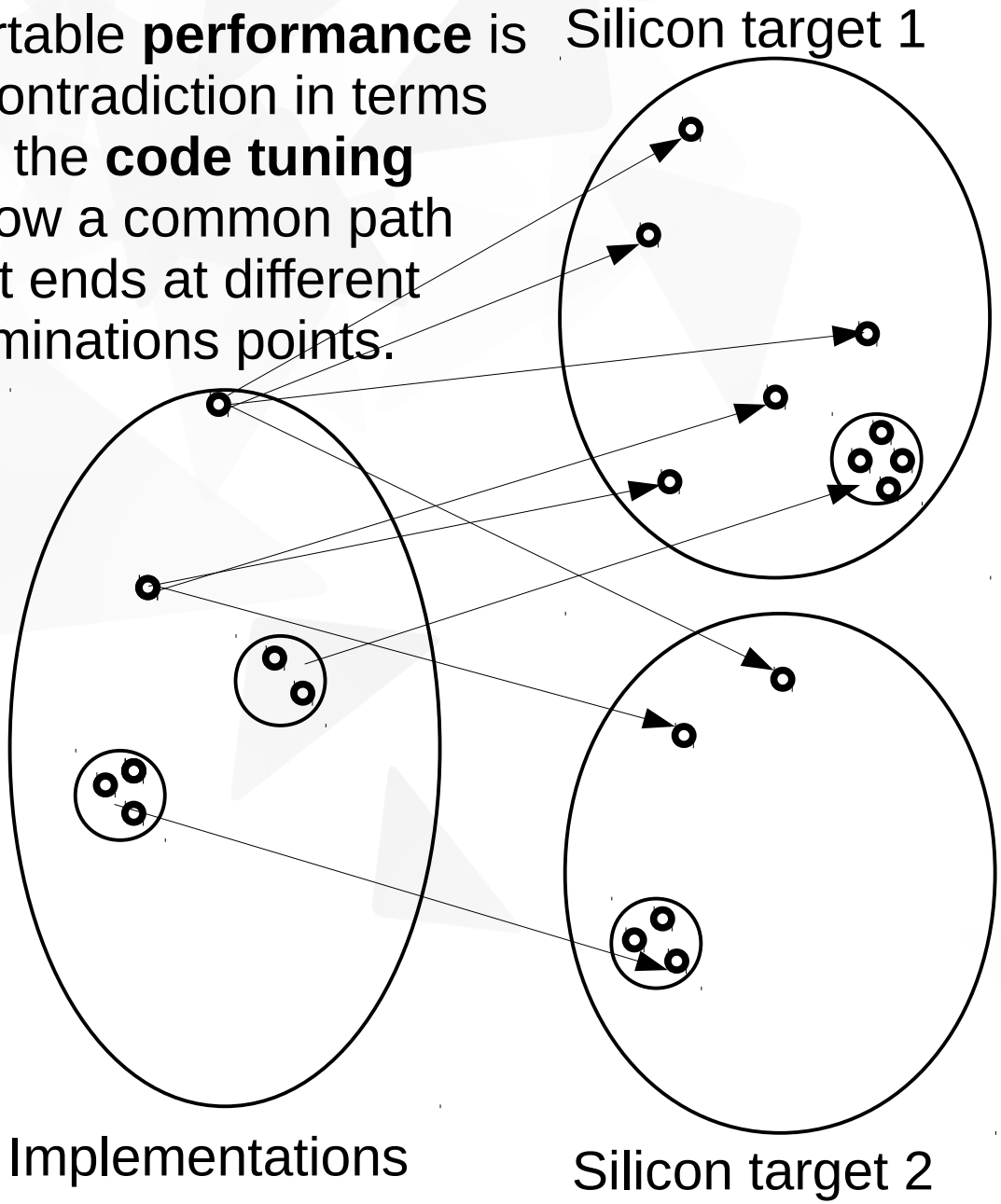


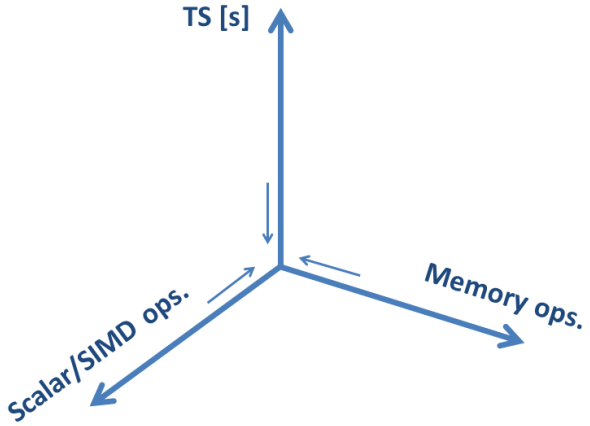
Memory footprint [Gb] after refactoring

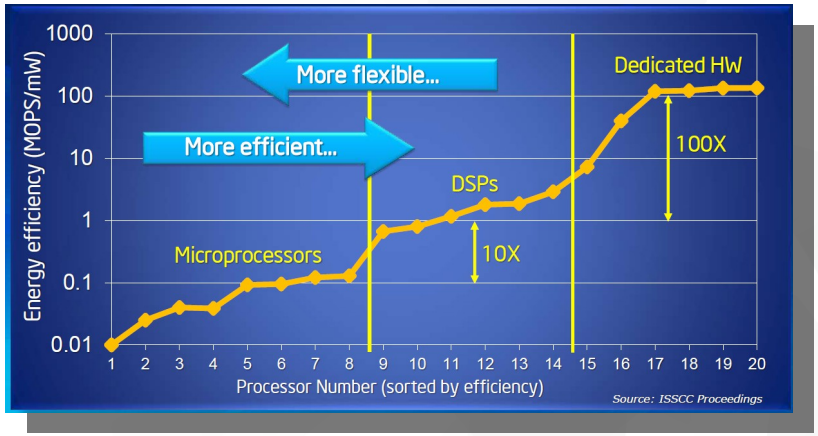


Performance target is to reach the infimum

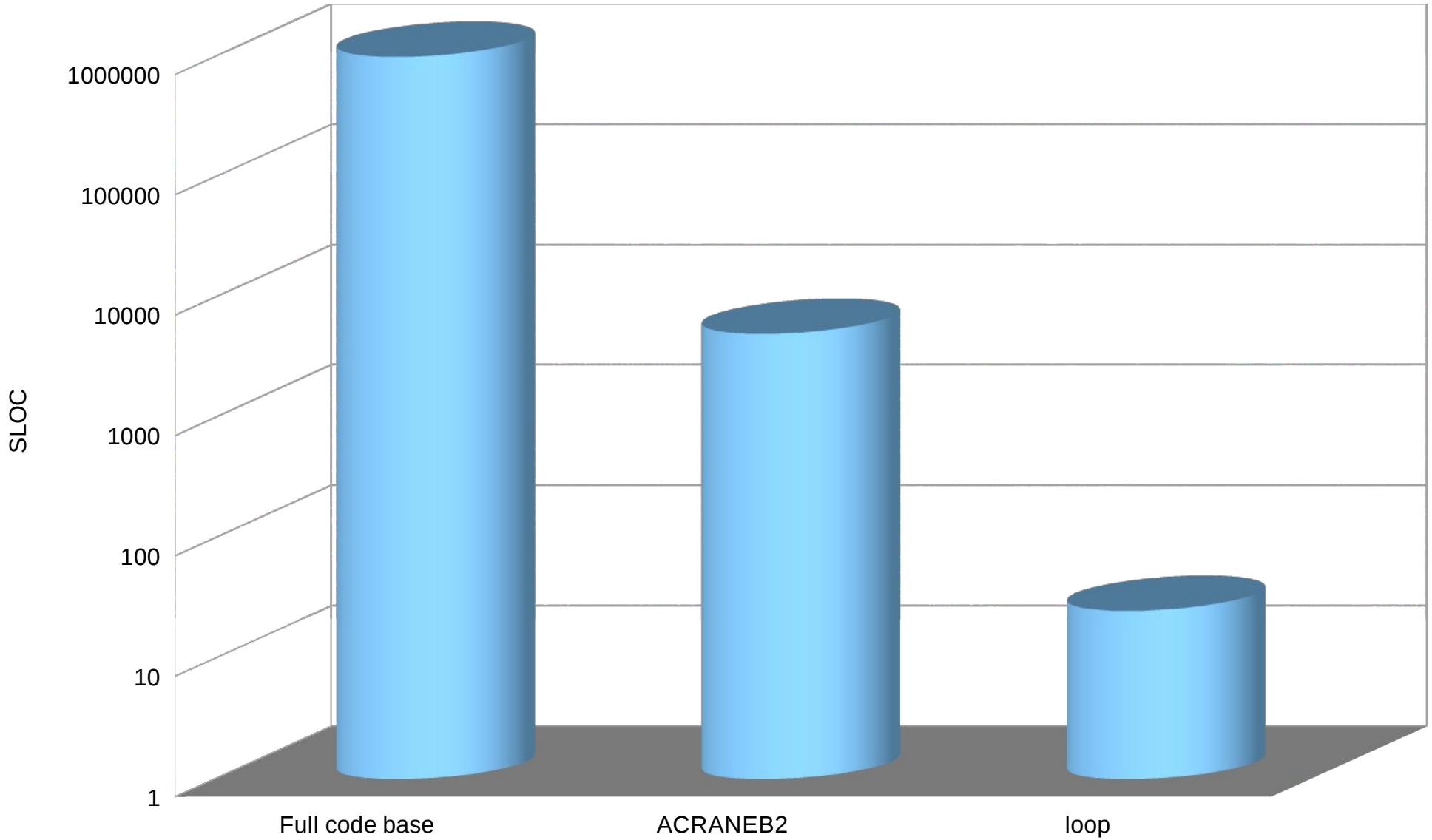
Portable **performance** is a contradiction in terms but the **code tuning** follow a common path that ends at different terminations points.







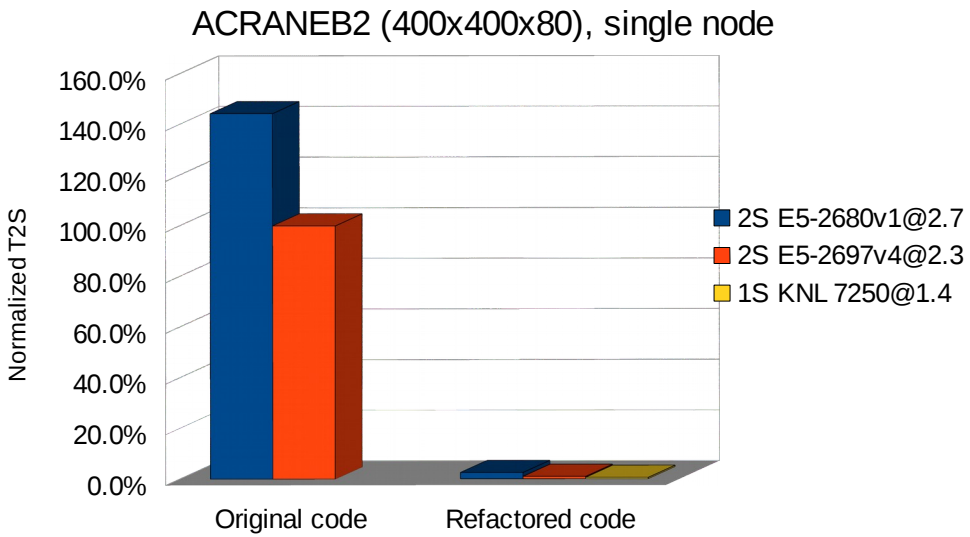
Legacy model → dwarf → kernel (PATTERNS!)

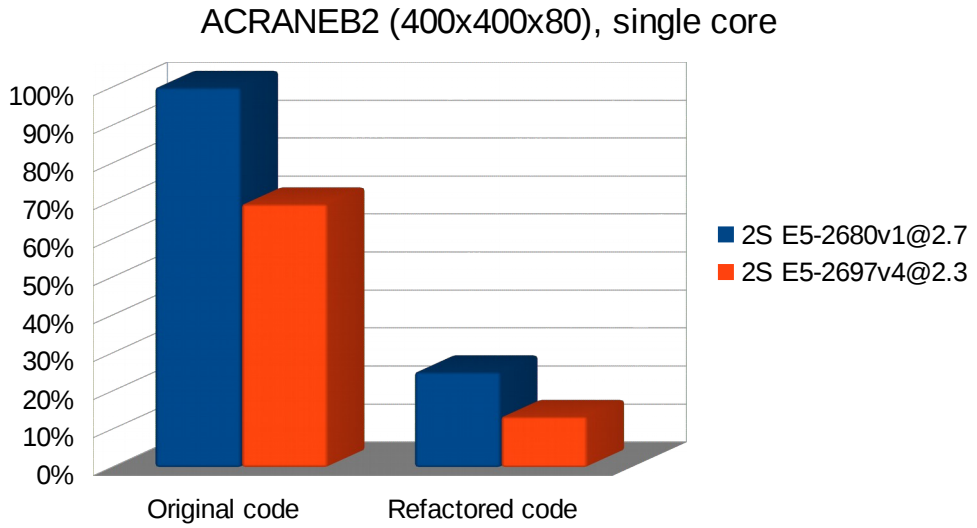


Investment in software vs hardware

Largest ACRANEB2 testcase (400x400x80) that the original code could fit into the 64Gb of RAM available on one node:

	Time-to-solution			Memory
Code	E5-2680v1@2.7	E5-2697v4@2.3	KNL 7250@1.4	E5-2697v4@2.3
Baseline	375%			
Version 0	144%	100%		100%
Refactored	2.58%	0.92%	0.52%	17.4%





Conclusion

- This experiment has shown that one can improve the common multicore performance of current physics (same pattern) in IFS/Harmonie by following these general steps:
  - Reduce implementation overhead (primarily memory operations)
  - Thread parallelize entire computation using a SPMD approach
  - re-SIMD vectorize all column loops by proper split of computations
- This refactoring of the code lead to even more efficient (both in time2solution and in Watt2solution) code on the high-end many-core architectures (Xeon Phi and GPUs).
- Important observations:
  - A (or parts of A) may not always have perfect fit on a given piece of silicon