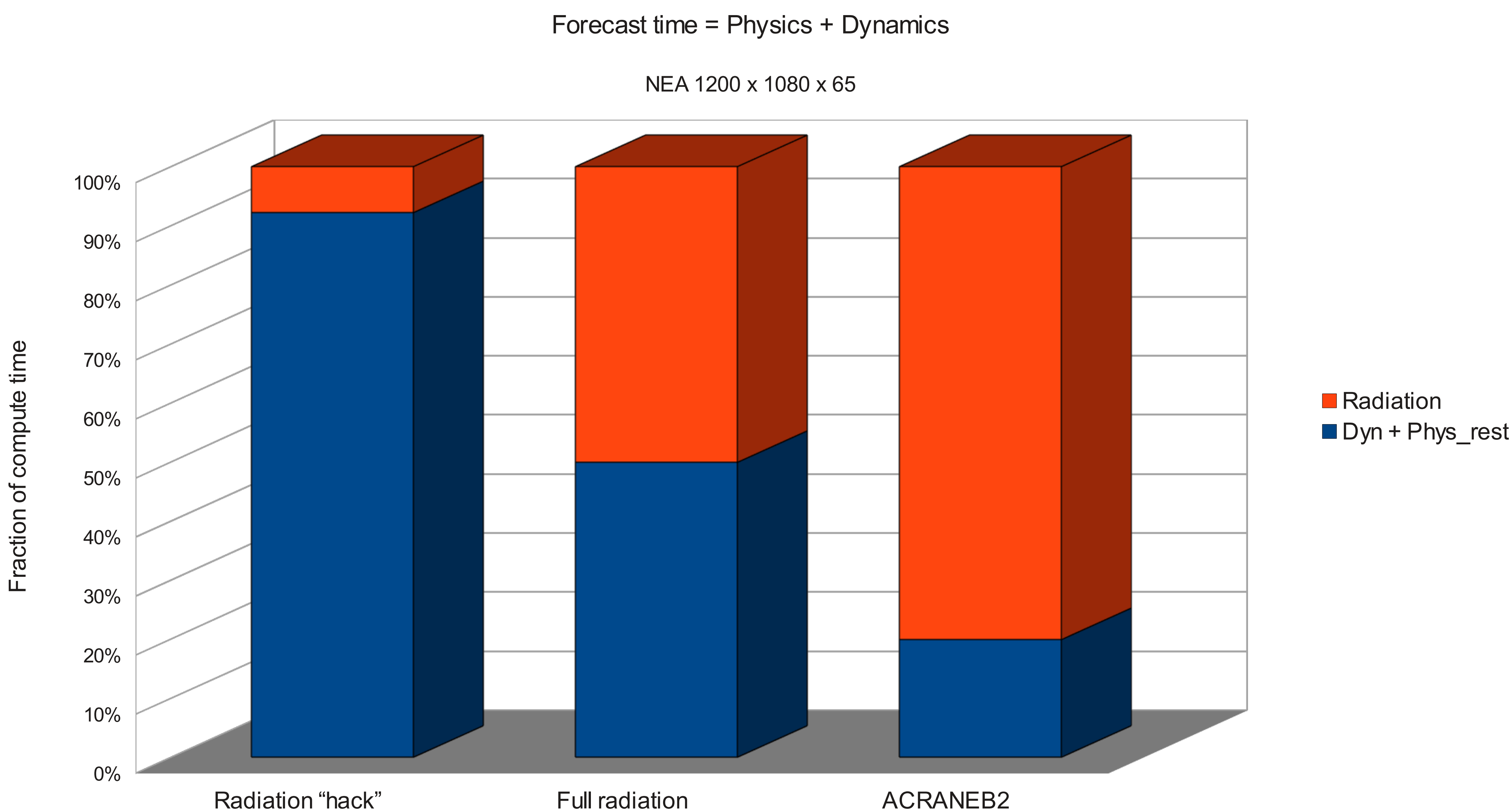
Recent progress on the ACRANEB2* dwarf from the ESCAPE project, part I

Per Berg and Jacob Weismann Poulsen
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)

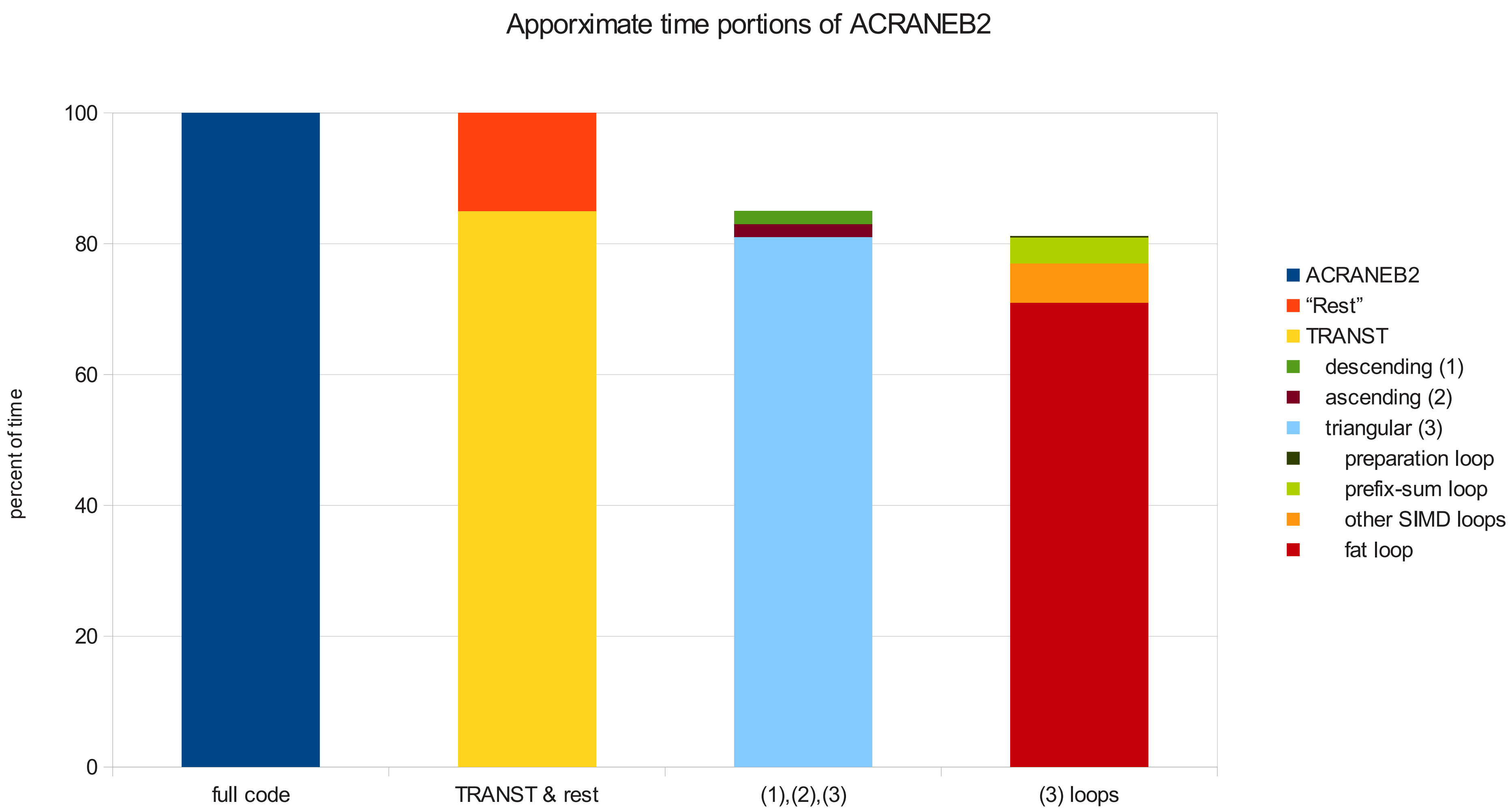
Motivation: Radiation might become the overall bottleneck in the future



Initial study (autumn 2016)

Baseline		case [s]
SLOC [lines]	5687	
Language state	F77,fixed-size, type-cast,...	
Technical state	OK	
Numerical state	4 digits	
Largest psize on 16Gb	200x200x80	
Largest psize on 64Gb	400x400x80	1653
Largest psize on 128Gb	600x600x80	
Target psize	1200x1080x65	

Continued studies (autumn 2016 – Feb 2017)



Split **TRANST** in three parts:

(1) Descending

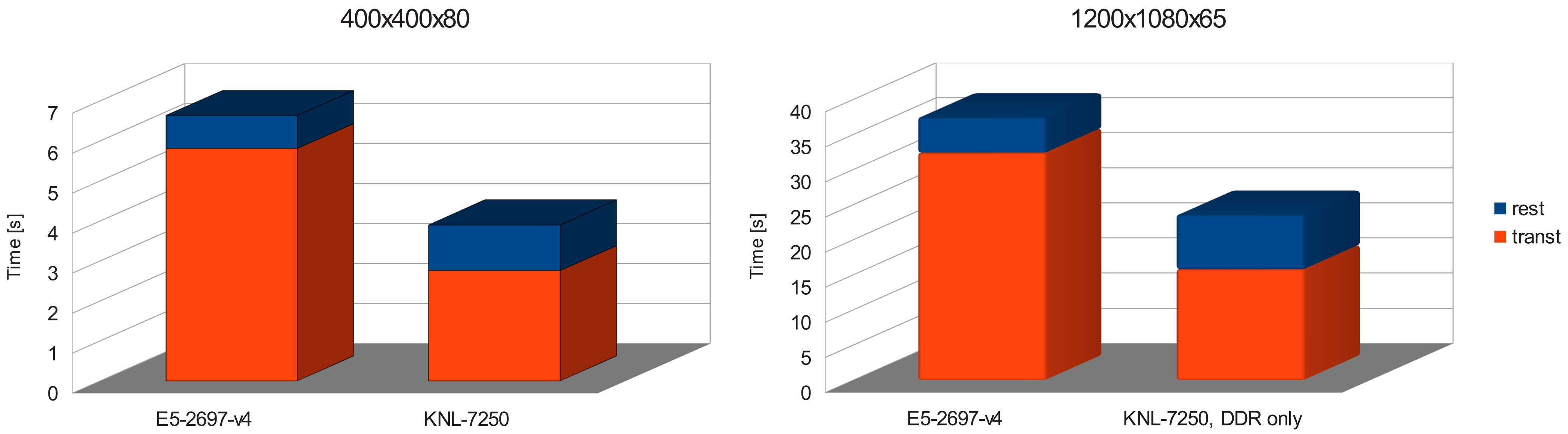
(2) Ascending

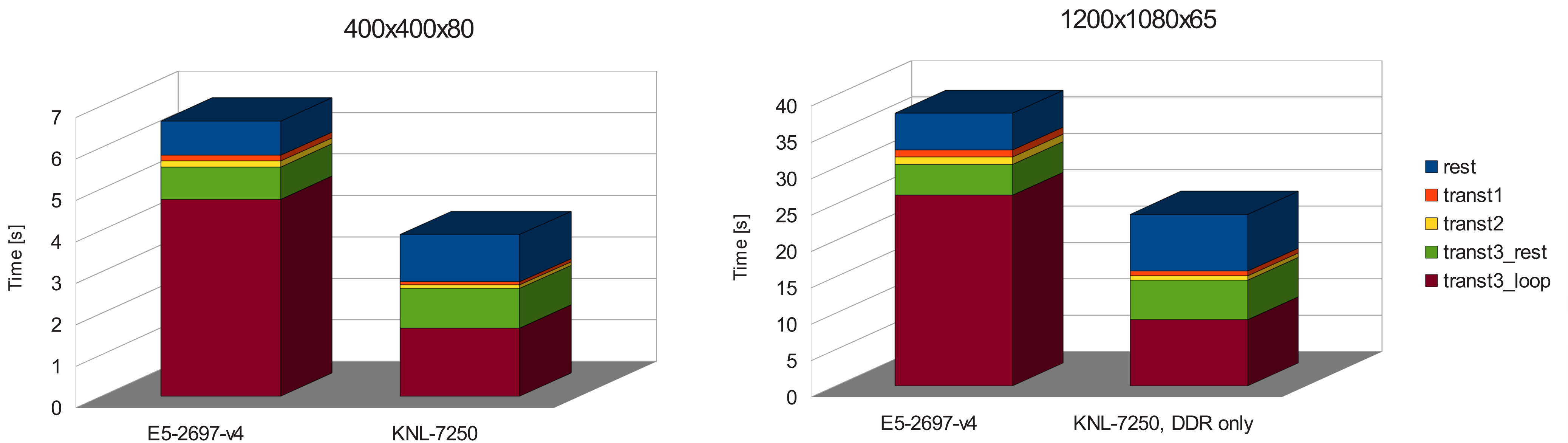
(3) Triangular

Triangular part is further split into:

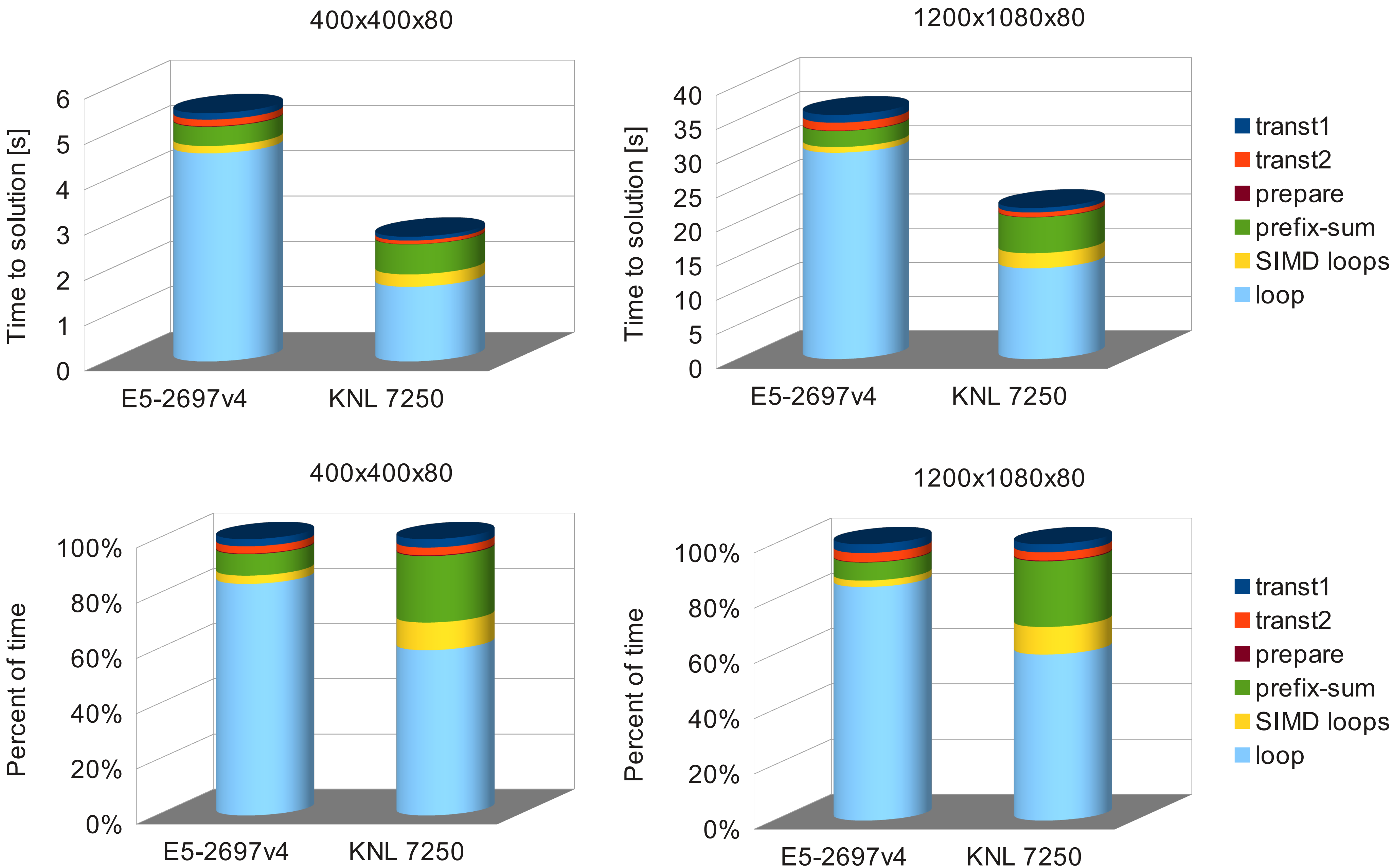
- Small preparation SIMD loop
- Prefix-sum: >100% BW on BDW, close to peak on KNL
- Other loops, mostly SIMD loops
- Focus on the **fat loop**

Complete **ACRANEB2** dwarf (small + large testcases)





Splitting of **TRANST** (small + large testcase)



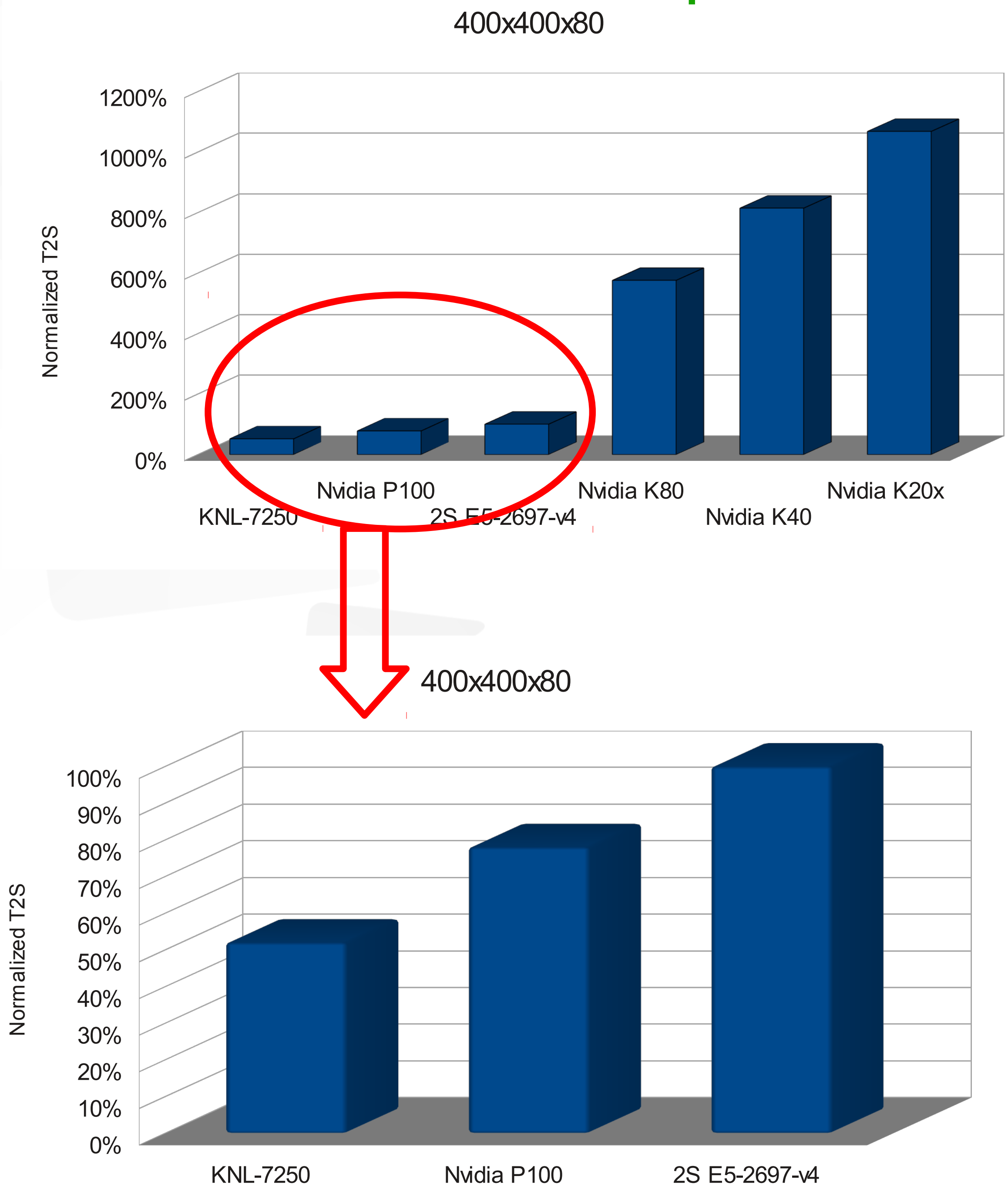
ESCAPE, preliminary results, March 2017

Zoom in on **fat loop** in **TRANST3**

ops	**#ops in loop**
POW	22
EXP	8
LOG	14
SQRT	18
DIV	48
MUL	308
ADD	454
MAX	24

- characterized by these operations (one iteration):
  - **Fat**: DP arithmetic intensity ~8-12 FLOPS/Byte
  - ~60–80% of **TRANST3** time in this **loop**
- 4 decimals reproduced in results depending on choice of math library and compiler flags
- We did not question the mathematical physics of the problem
  - Did only a few algebraic re-writes reducing #DIV and #SQRT
  - **maintained same results** (at least 9 decimals)
- Performance is pretty good, e.g. sustaining >1 TFLOP/s with “MKL svml la” on Knights Landing, **~48% of peak**

Zoom on **TRANST** – performance across platforms

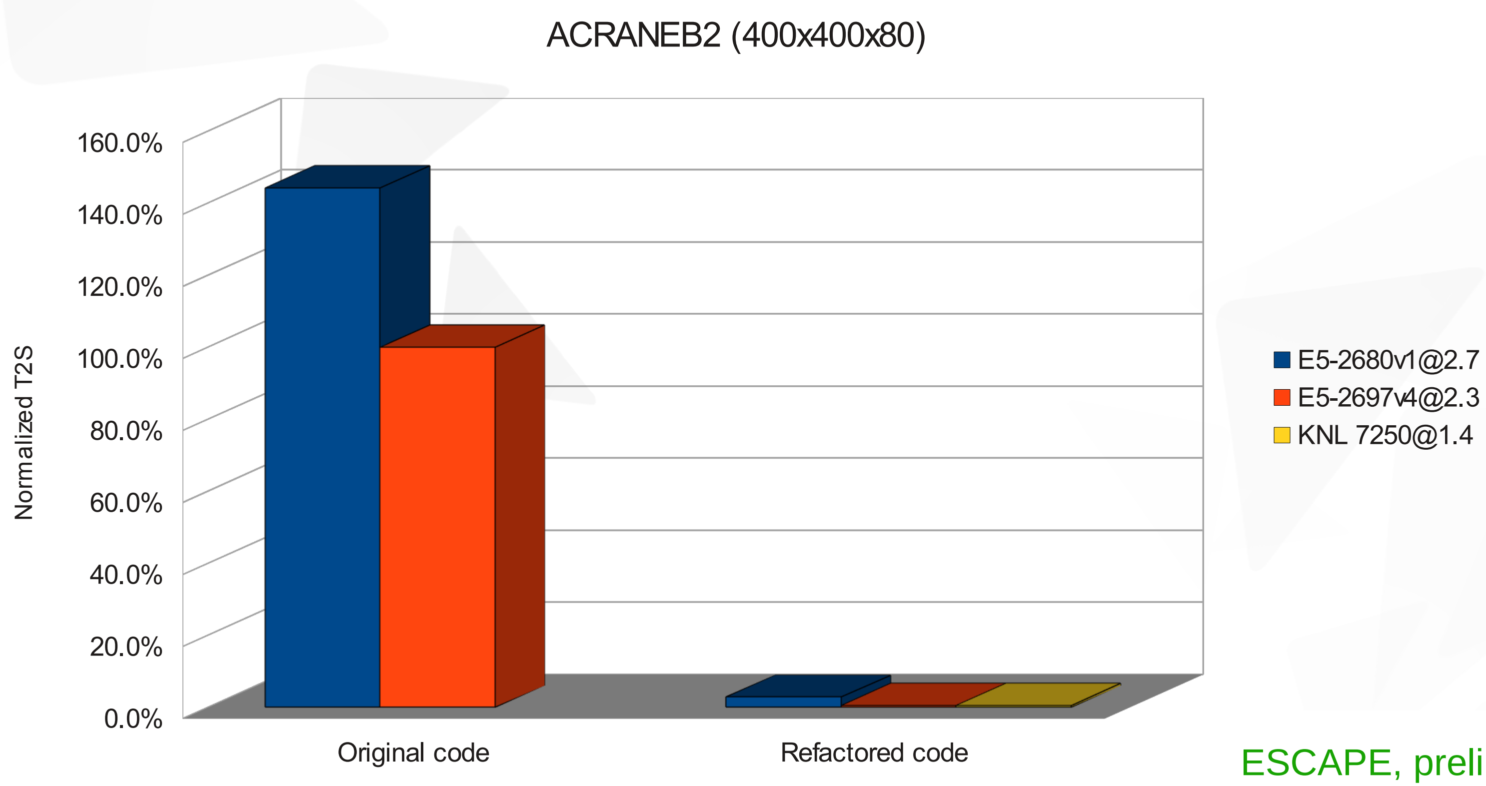


- Cross-compare single KNL and single GPU
  against
  2S E5-2697-v4, 72 threads
- No accounting of PCI communication, i.e. compute time only on the GPUs
- This is best performance on target at hand and hence ***NOT portable performance*** across the 3 platforms

Investment in software vs hardware

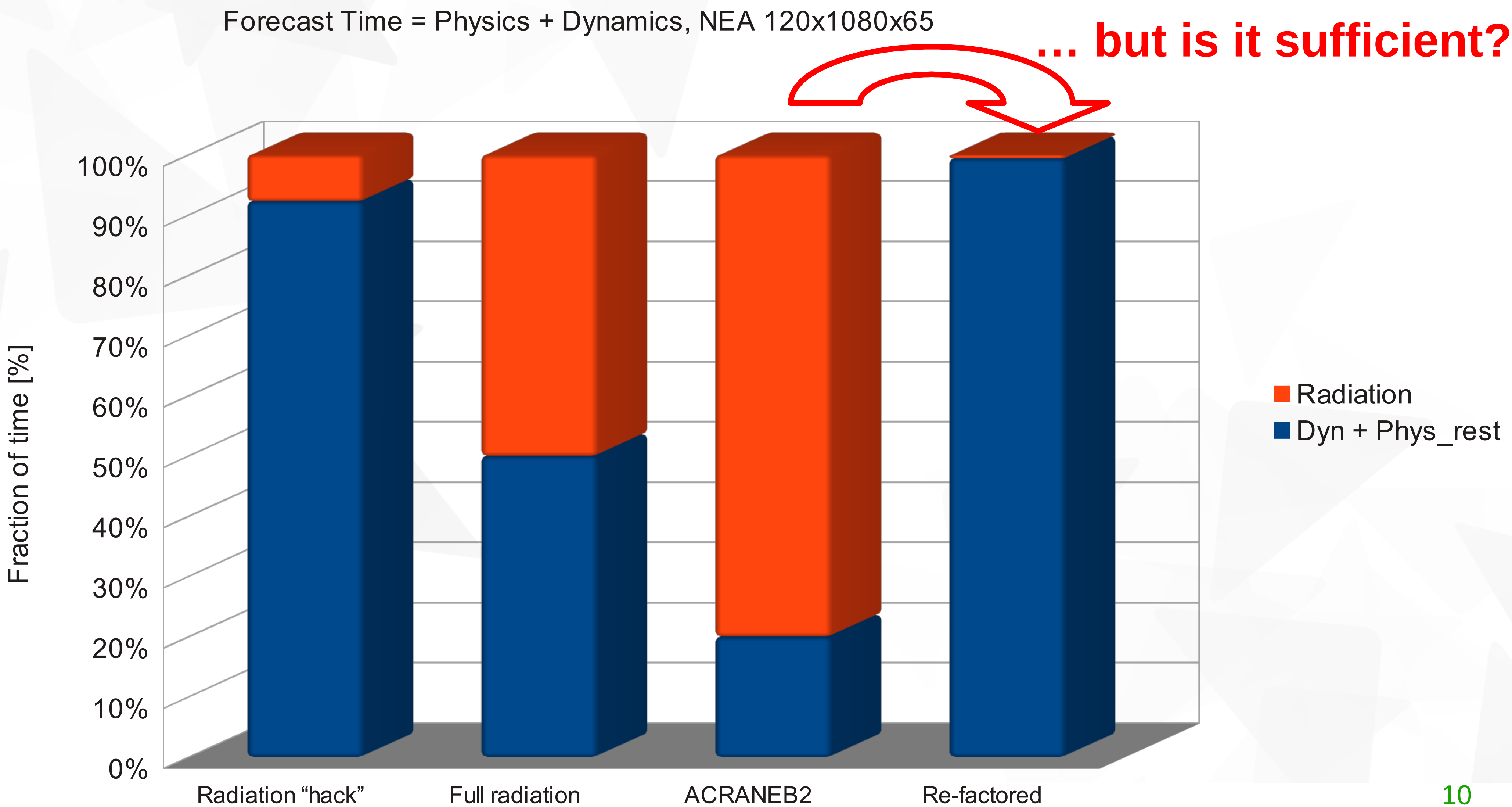
- Largest ACRANEB2 testcase (400x400x80) that the original code could fit into the 64Gb of RAM available on one node:

	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	2.87%	0.85%	0.54%		17.4%



Motivation: Radiation might become the overall bottleneck in the future...

Conclusion: ... but software re-factoring allows us to do much more physics under the fixed constrains on time-to-solution and hardware investment.

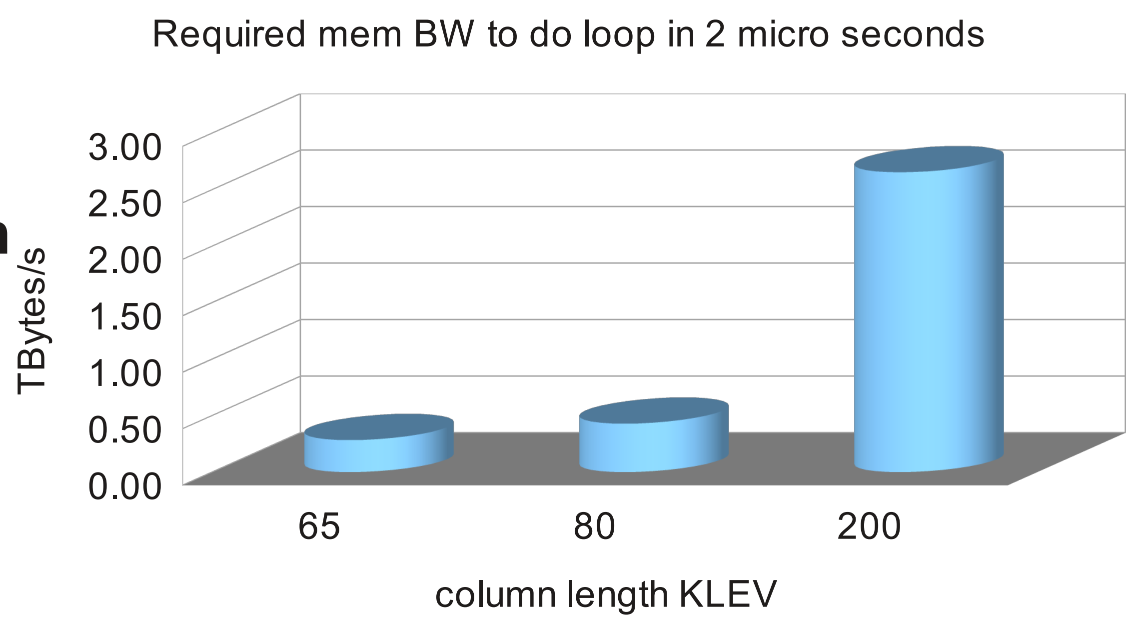


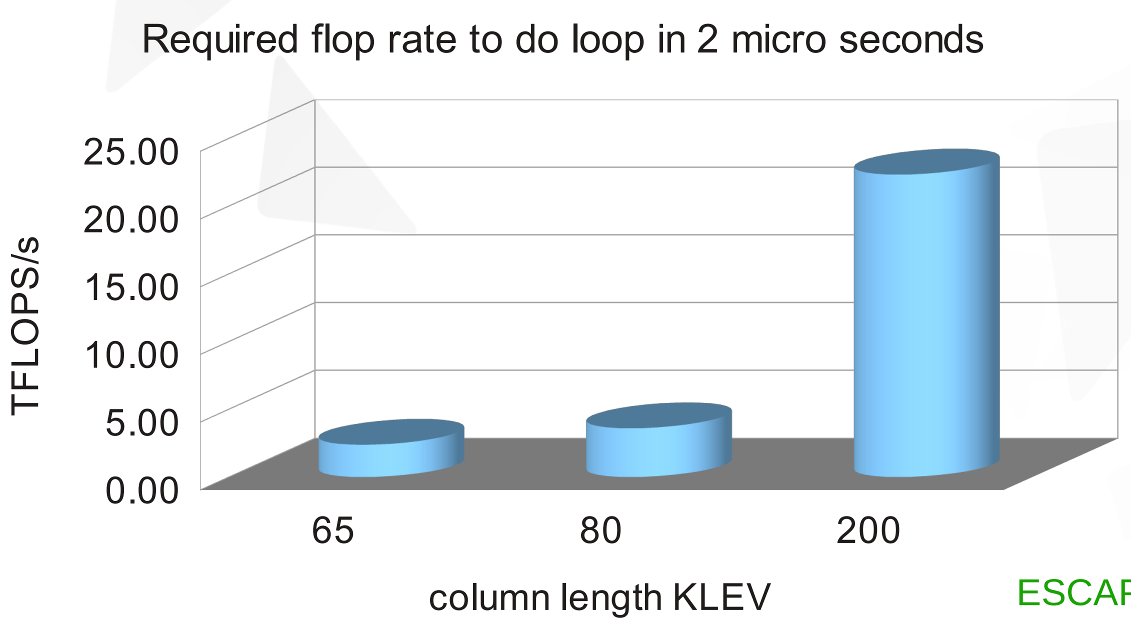
Perspectives – an example

ESCAPE “Performance Metrics” by Andreas Müller, ECMWF:

Goal for 2025: 5 km model using 50 secs/1000 time steps on radiation

- Translates into requirements on radiation implementation: **~2.5 micro seconds per column**
- With ACRANEB2, the transt3 fat loop alone must sustain
  - **> 20 TFLOPS/s**
  - **> 2.5 TByte/s**





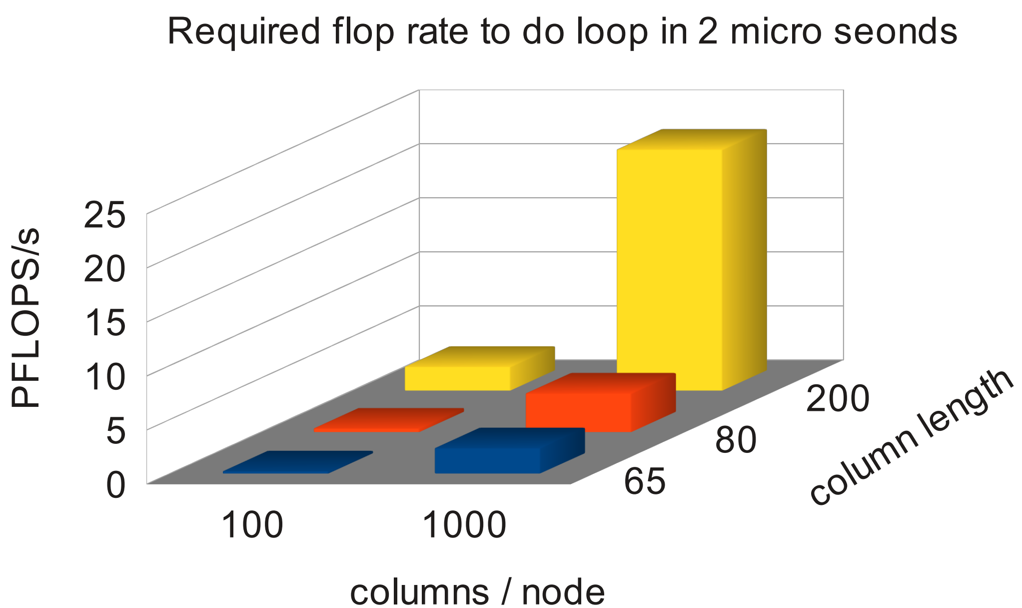
Perspectives (2)

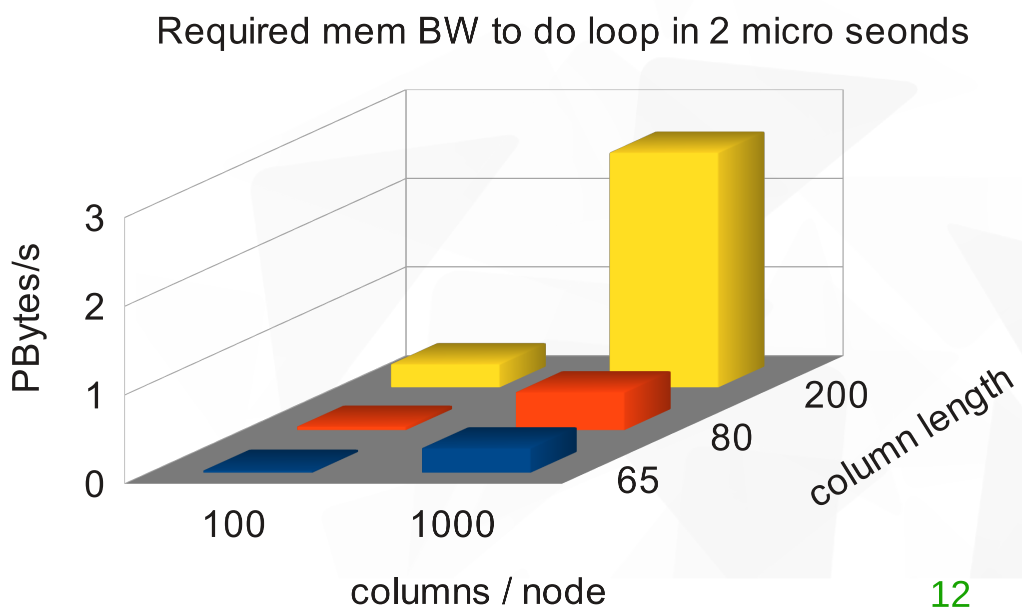
Assuming algorithm (the operations) is fixed, what is the requirement to HW given fixed constraint on time-to-solution ?

- Strong scaling on dynamics will impose constrains on min. #columns/node
- Exa-scale projections: **~1 TBytes/s** and **~2 TFLOPS/s** per node with **~1000 cores/node**
- At least 1 column per core or thread i.e. **columns/node > 1000**.

KLEV	PBytes/s	PFLOPS/s
65	0.28	2.4
80	0.43	3.6
200	2.7	22.3

@1000 columns/node

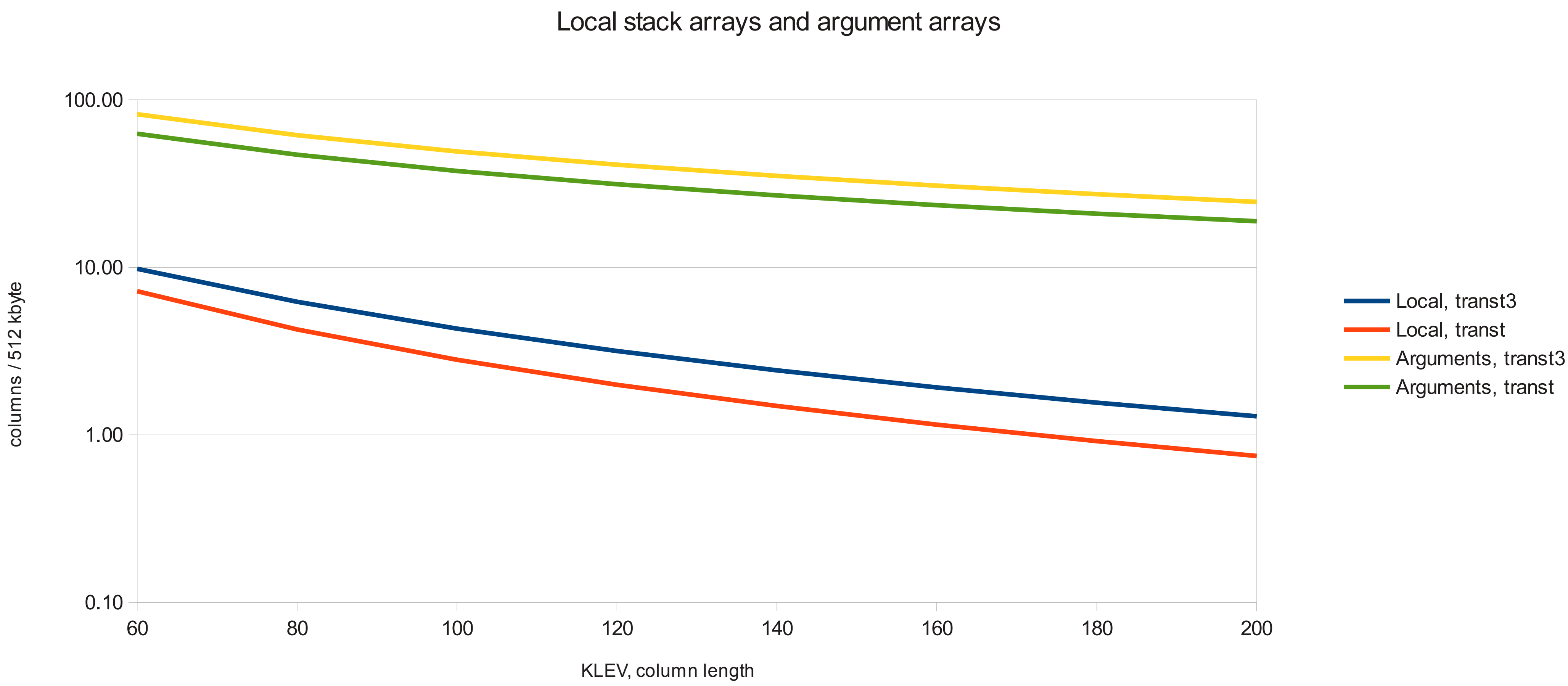




Continued studies (autumn 2016 – Feb 2017)

Recap on costs:

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, preliminary results, February 2017

Phenomenological modelling, metric1: flops

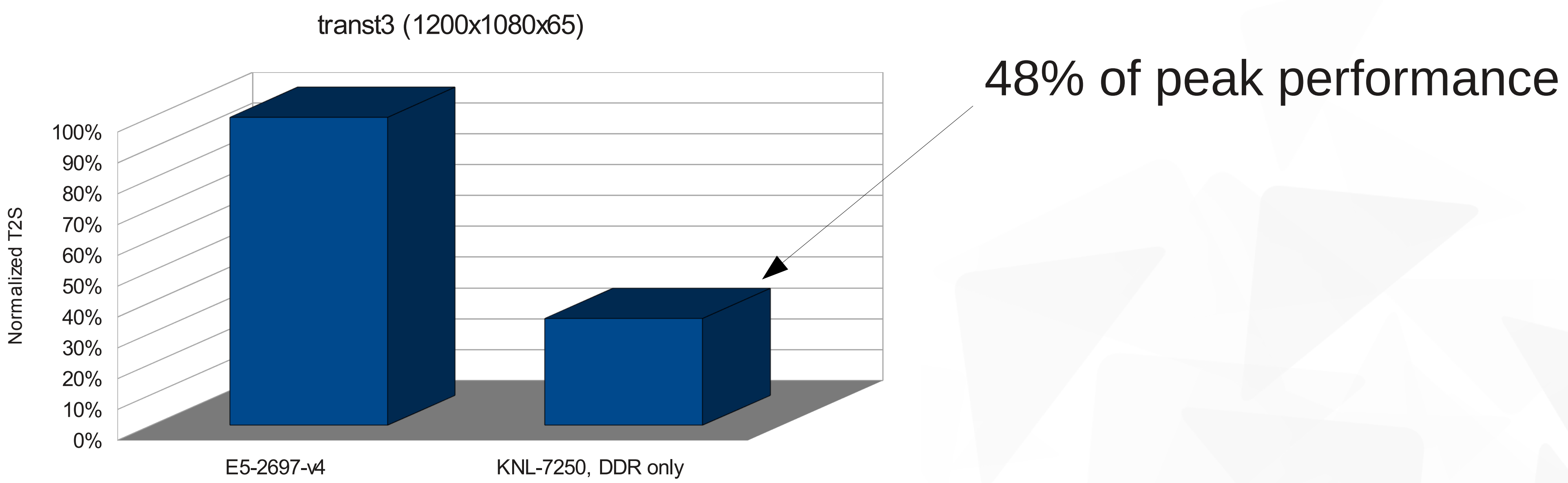
- Use tools (craypat, advisor, sde) to count flops for the math operations

ARCH		craypat				sde-craypat			
BDW	**ops**	**mkl**	**mkl svml ha**	**mkl svml la**	**mkl svml ep**	**mkl**	**mkl svml ha**	**mkl svml la**	**mkl svml ep**
	max	1	1	1	1	0	0	0	0
	fma	2	2	2	2	0	0	0	0
	div	1	1	1	1	0	0	0	0
	sqrt	1	1	1	1	0	0	0	0
	exp	19	20	14	10	0	1	0	0
	log	25	21	16	12	-1	2	3	3
	pow	55	64	47	21	0	0	2	1
KNL	*max*	2	2	2	2	-1	-1	-1	-1
	fma	2	2	2	2	0	0	0	0
	div	2	16	8	8	-1	-15	-2	-2
	sqrt	2	15	15	15	-1	-1	-1	-1
	exp	88	17	16	11	-69	6	6	2
	log	66	29	22	17	-38	5	6	2
	pow	201	77	72	41	-146	1	0	-1

Phenomenological modelling, metric1: flops

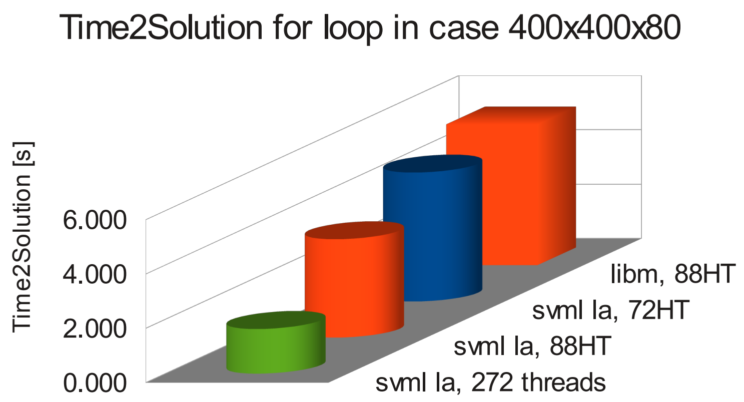
- Cross-compare model with measurements, reasonable results but only useful for coarse grained projections, deviations in percent.

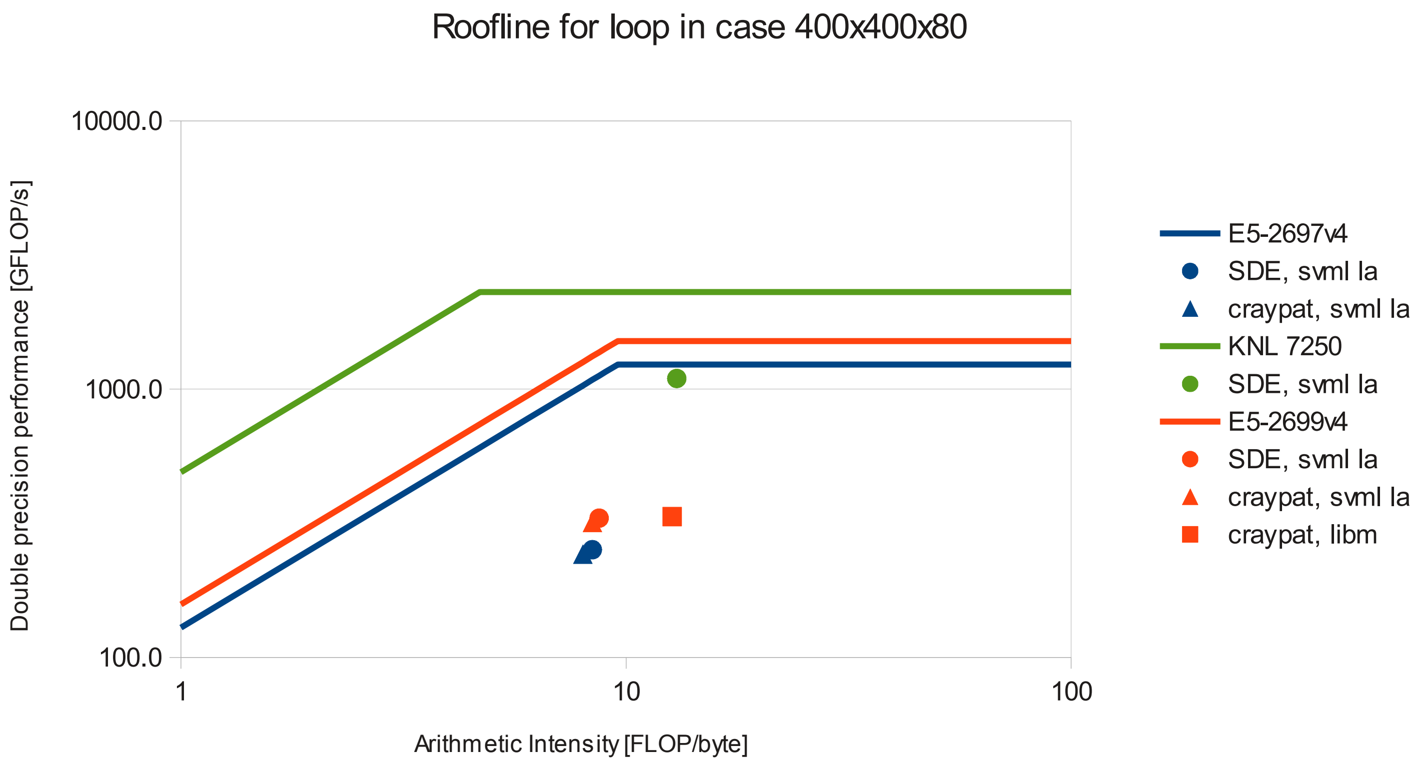
HW	Tool	mkl	svml ha	svml la	svml ep
E5-2697v4	sde	-8.1	-1.4	-3.1	-7.2
E5-2699v4	craypat	-6.6	0.0	-1.9	-5.9
KNL-7250	sde	-8.0	-3.6	-3.0	-3.1
KNL-7210	craypat	-4.7	-1.6	-1.5	-4.5



Roofline, phenomenological, metric: flops

- Alas, limited insight into the real performance bottlenecks, flops does not represent a good metric for Pmax in this particular case.





Phenomenological modelling, metric2: cycles

- Benchmark or lookup (a few architectures are documented):

	v4-ha	v4-la	v4-ep	KNL-ha	KNL-la	KNL-ep
Add	0.47	0.47	0.47	0.47	0.47	0.47
Mul	0.47	0.47	0.47	0.47	0.47	0.47
Div	4.19	3.13	2.45	1.95	1.42	1.16
Sqrt	4.07	3.97	3.58	1.8	1.51	1.08
Exp	5.56	3.72	3.35	2.53	2.23	1.78
Log10	6.89	5.7	5.19	3.42	2.99	2.26
Pow	20.88	11.92	10.49	9.87	7.94	4.85

- But how do we find the critical path for the remaining operations and how do we handle the “complex math”, assume no overlap or ?