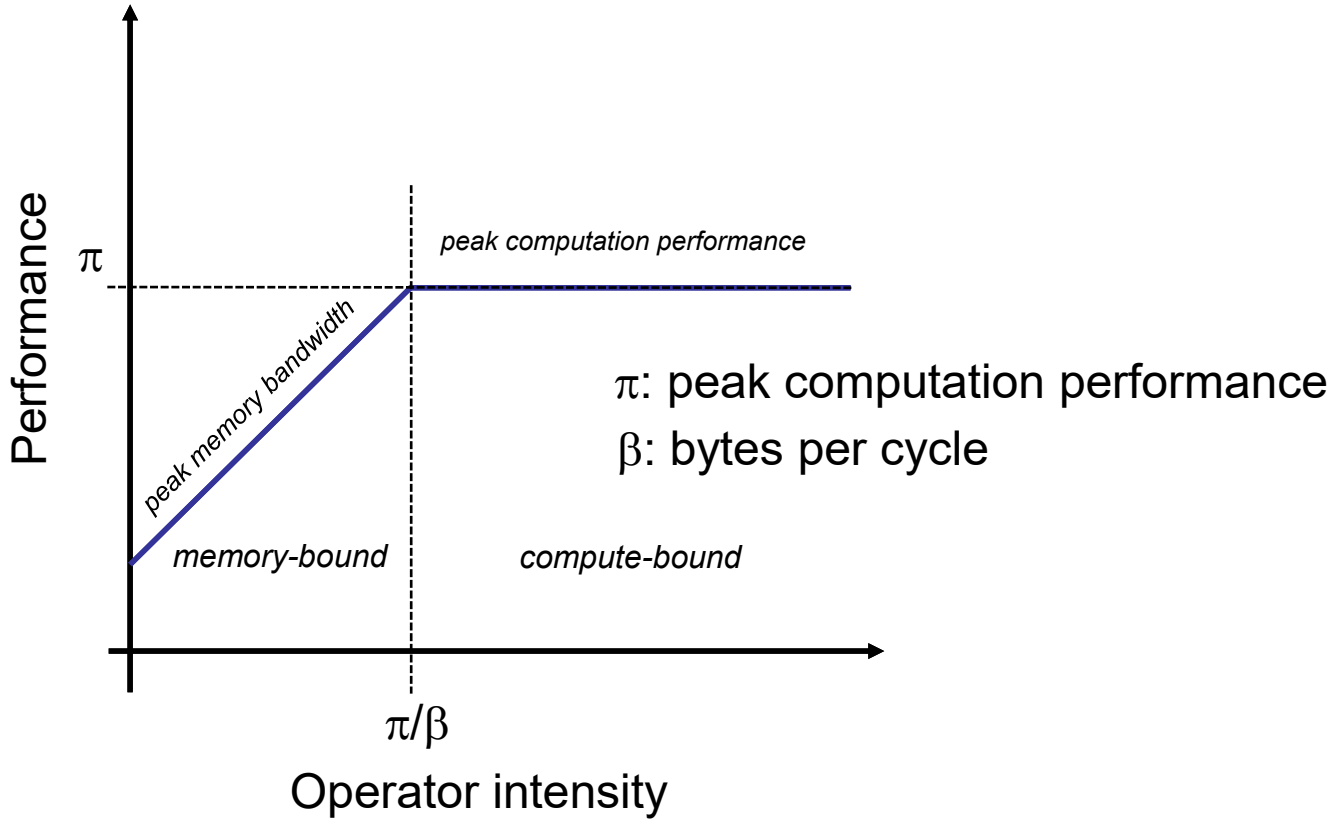
The Roofline Model

Msc. in Informatics and Computing Engineering (M.EIC), 2nd Year

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Roofline Model

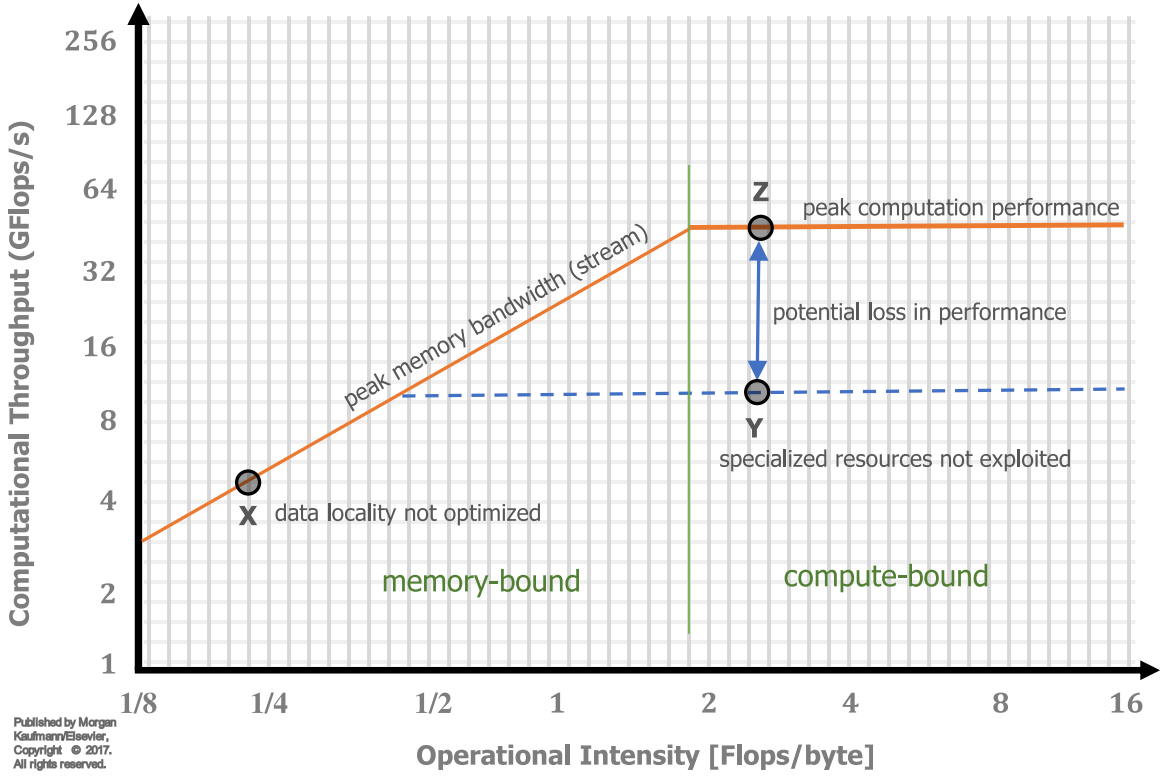


- Log-scale for axis
- Performance (e.g., GFlops/sec, Flops/cycle)
- Operator/Arithmetic Intensity (e.g., Flops/byte)

S. Williams, **"The roofline model",** book chapter in Performance Tuning of Scientific Applications, Chapman & Hall/CRC Computational Science, CRC Press, Editor(s): David H. Bailey, Robert F. Lucas, Samuel Williams, Nov. 2010, pp. 195-215. https://escholarship.org/content/qt0qs3s709/qt0qs3s709.pdf

Samuel Williams, Andrew Waterman, and David Patterson. 2009. "**Roofline: an insightful visual performance model for multicore architectures,**" Commun. ACM 52, 4 (April 2009), 65–76. DOI: https://doi.org/10.1145/1498765.1498785

Roofline Model

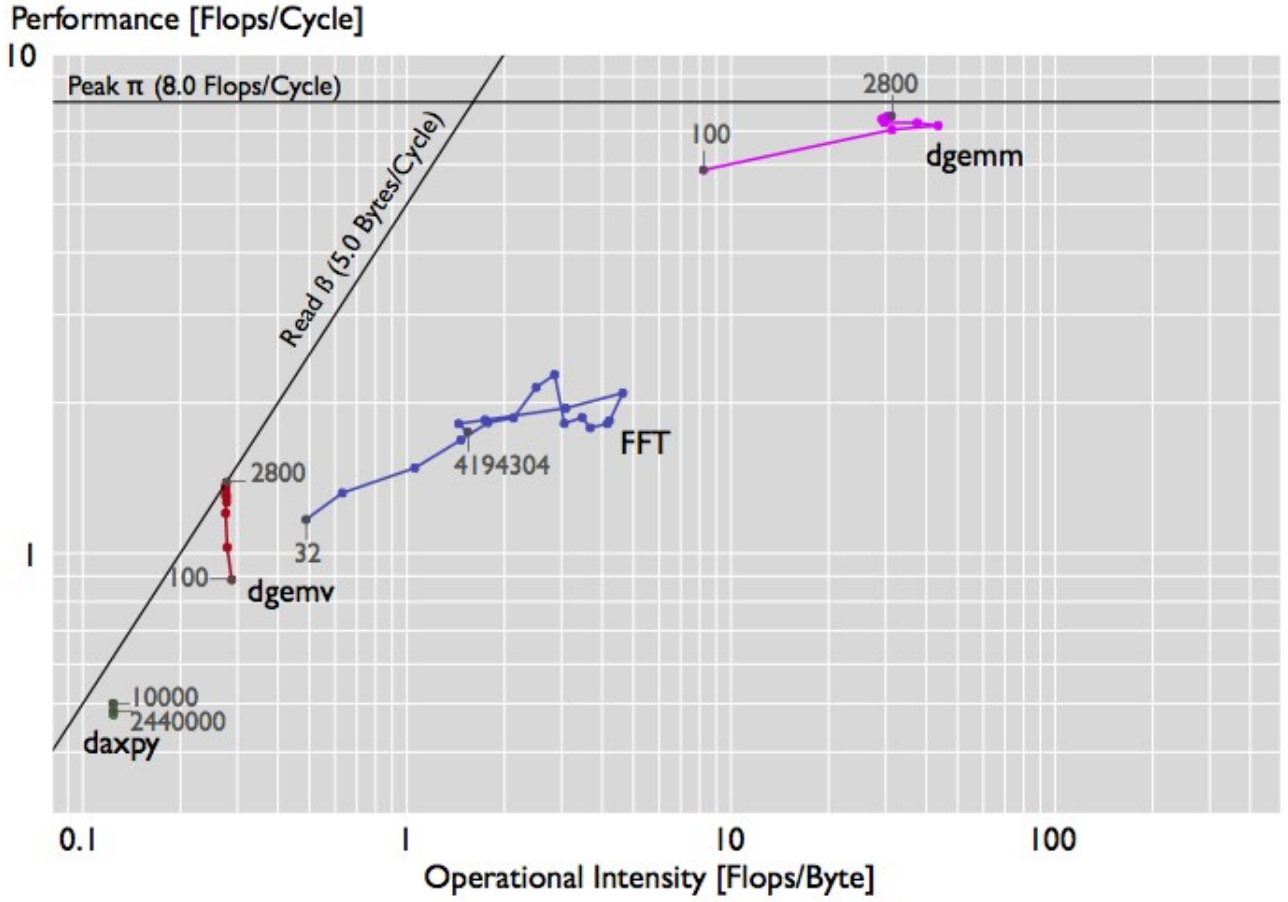


- Log-scale for axis
- Performance (e.g., GFlops/sec, Flops/cycle)
- Operator/Arithmetic Intensity (e.g., Flops/byte)

S. Williams, **"The roofline model",** book chapter in Performance Tuning of Scientific Applications, Chapman & Hall/CRC Computational Science, CRC Press, Editor(s): David H. Bailey, Robert F. Lucas, Samuel Williams, Nov. 2010, pp. 195-215. https://escholarship.org/content/qt0qs3s709/qt0qs3s709.pdf

Samuel Williams, Andrew Waterman, and David Patterson. 2009. "**Roofline: an insightful visual performance model for multicore architectures,**" Commun. ACM 52, 4 (April 2009), 65–76. DOI: https://doi.org/10.1145/1498765.1498785

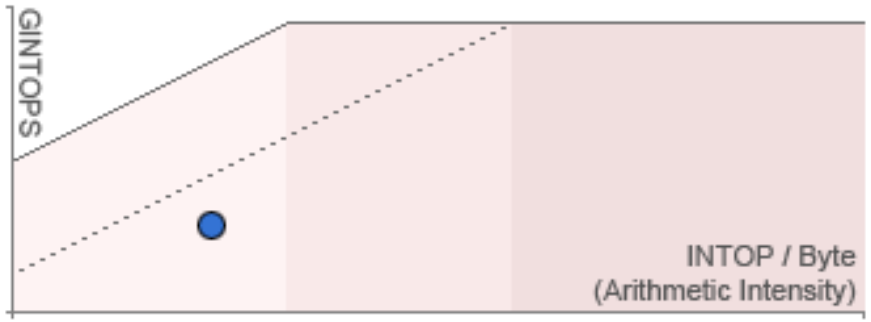
Roofline Model: Example

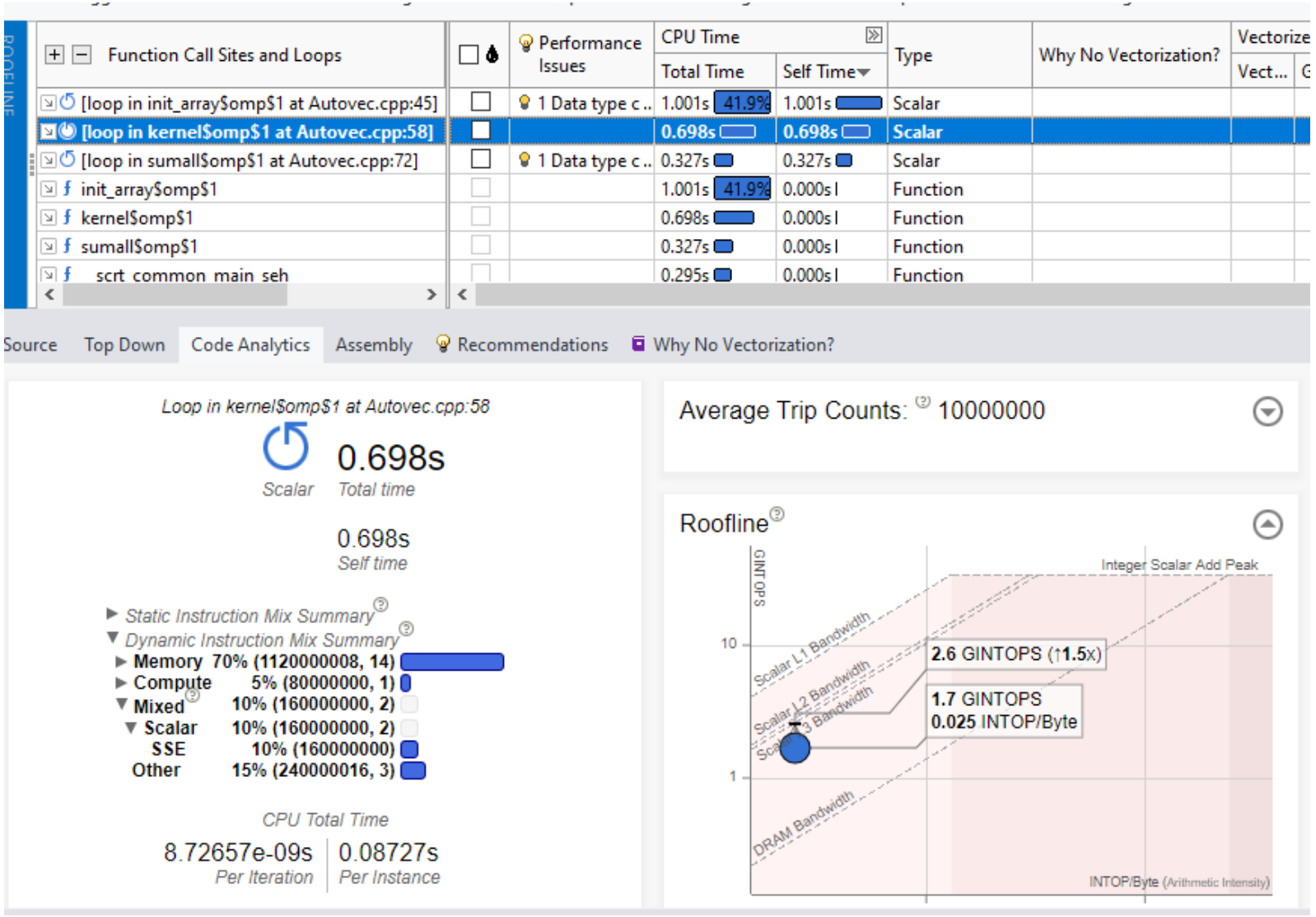


From: G. Ofenbeck, R. Steinmann, V. Caparros, D. G. Spampinato and M. Püschel**, "Applying the roofline model,"** *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2014, pp. 76-85, doi: https://doi.org/10.1109/ISPASS.2014.6844463.

Roofline model by Intel Advisor

- autovec example





Intel® Advisor

- Intel® Advisor Introduction | Intel Software
  - https://www.youtube.com/watch?v=K4S5-q6ydi8 [6:13]
- An Introduction to the Roofline Feature in Intel® Advisor 2017
  - Roofline Analysis in Intel Advisor 2017: https://youtu.be/h2QEM1HpFgg
- Introduction to Intel® Advisor: Vectorization Workflow
- Intel® Advisor Tutorial: Use the Automated Roofline Chart to Make Optimization Decisions
  - https://www.intel.com/content/www/us/en/develop/documentation/advisor-tutorial-roofline/top.html
  - https://github.com/oneapi-src/oneAPI-samples

https://www.intel.com/content/www/us/en/developer/tools/oneapi/advisor.html#gs.ebqvim

[VIDEO: 17'] https://www.intel.com/content/www/us/en/developer/videos/roofline-analysis-in-intel-advisor-2017.html

https://www.intel.com/content/www/us/en/develop/documentation/advisor-user-guide/top.html

https://www.intel.com/content/www/us/en/develop/documentation/advisor-tutorial-roofline/top.html

https://www.intel.com/content/www/us/en/developer/videos/introduction-to-intel-advisor-vectorization-workflow.html

EXERCISES FROM LABORATORY SCRIPT #2-II

- Two target machine models:
  - A: simple machine
    - 1 load/store unit
    - 1 arithmetic unit for add/sub/mul/mac
  - B: machine w/ SIMD and FMA units
    - Vector length of 512 bits (64 bytes)
    - Simultaneously load/store of 16 floats (memory contiguous)
    - Simultaneously load/store of 8 doubles (memory contiguous)
    - Simultaneously 16 float SIMD/FMA operations - from vector registers to vector register
    - Simultaneously 8 double SIMD/FMA operations - from vector registers to vector register

Target machine models

- Main characteristics of the two Machine Models A and B
- Both machines without pipelining units

		Number of clock cycles (#ccs)					
		Machine A and B			Machine B		
Data type	**Bytes**	**mul/add/sub**	**mac**	**Load/store**	**Load/store (1 vector lane)**	**SIMD**	**FMA**
float (single-precision floating-point)	4	1	1	1	1	1	1
double (double-precision floating-point)	8	2	3	1	1	2	3

Target machine models

- Roofline model bounds

		Machine A and B (not considering SIMD/FMA and vector registers)		Machine B	
Data type	Bytes	β (bytes/cycle)	π (FLOPS/cycle)	β (bytes/cycle)	π (FLOPS/cycle)
float (single-precision floating-point)	4	8	2	64	32
double (double-precision floating-point)	8	4	1.5	64	5.33

Code examples

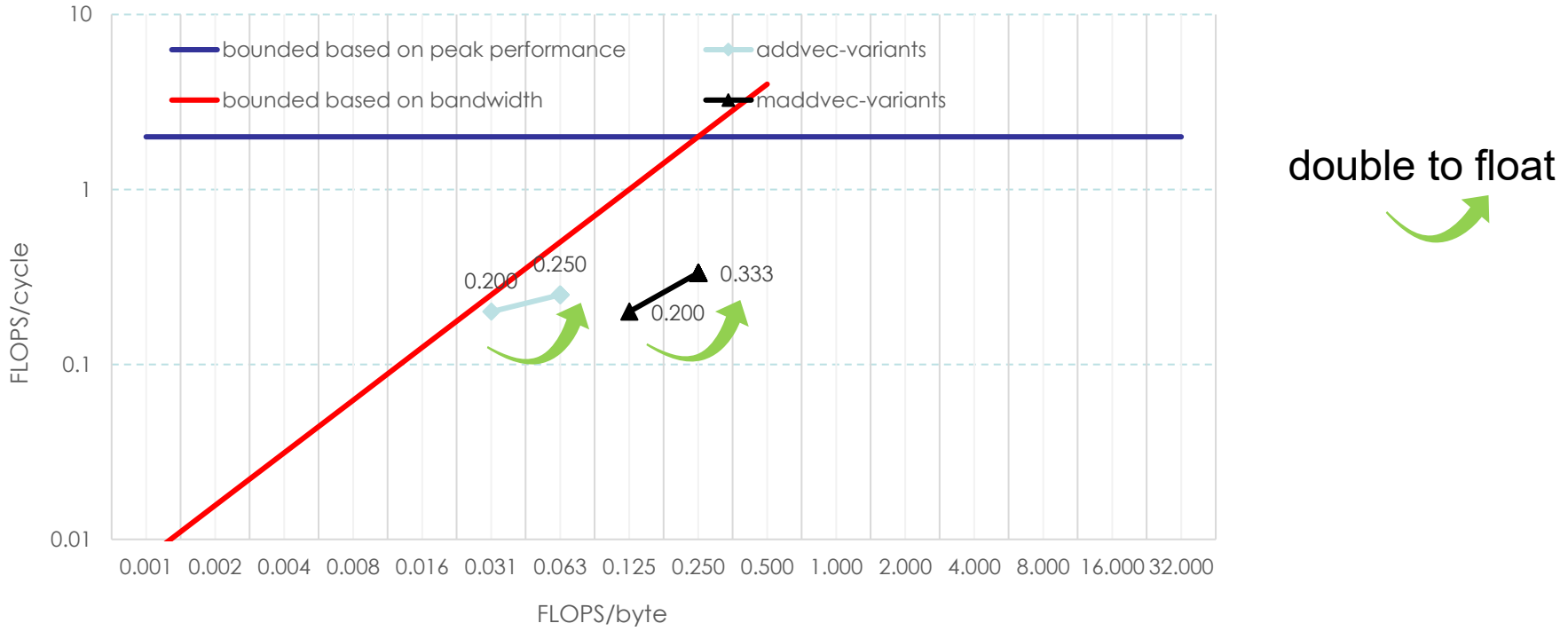
- Two simple code examples, considering float and double data types, and N=2048:

```
addvec    for (int i = 0; i < N; i++) {
              C[i] = A[i] + B[i];
          }

maddvec   for (int i = 0; i < N; i++) {
              C[i] = c*A[i] + d;
          }
```

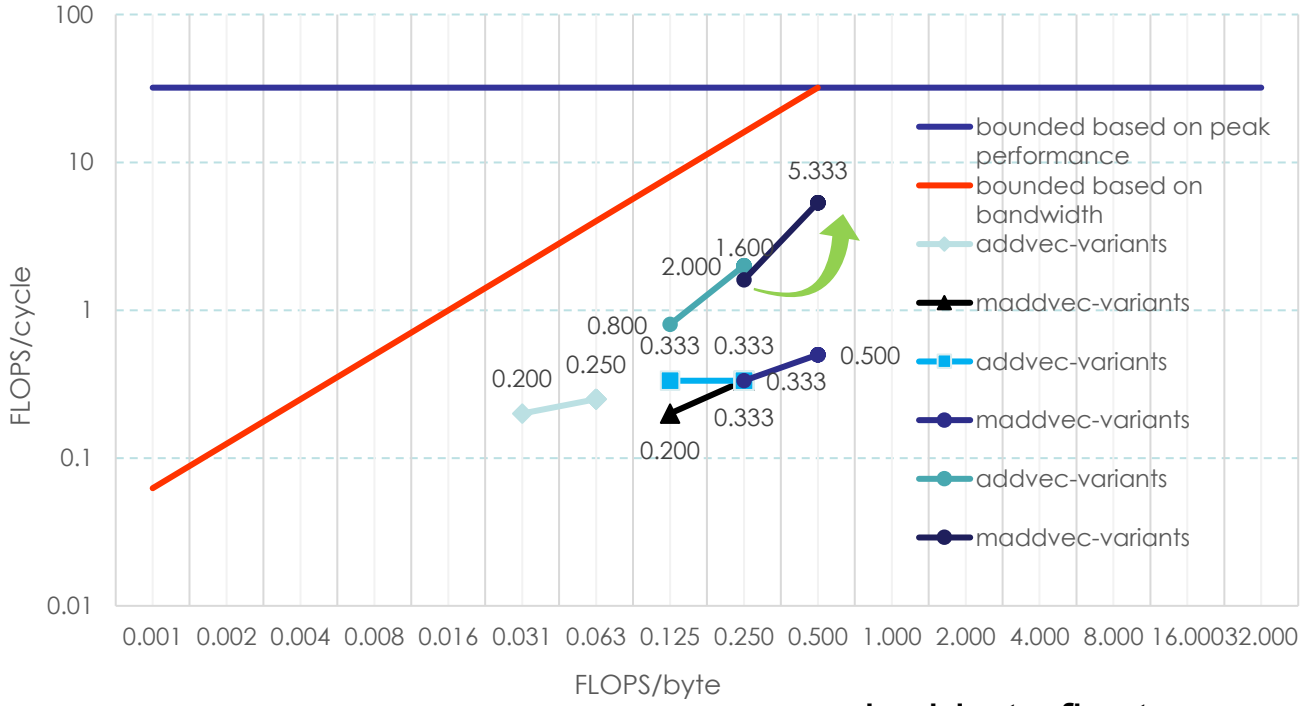
Roofline model

- *addvec* and *maddvec* examples in Machine A
- Bounds only show lines for single-precision floating-point



Roofline model

- *addvec* and *maddvec* examples in Machine B
- Comparison with SWP and loop vectorization using SIMD vs FMA
- Bounds only show lines for single-precision floating-point

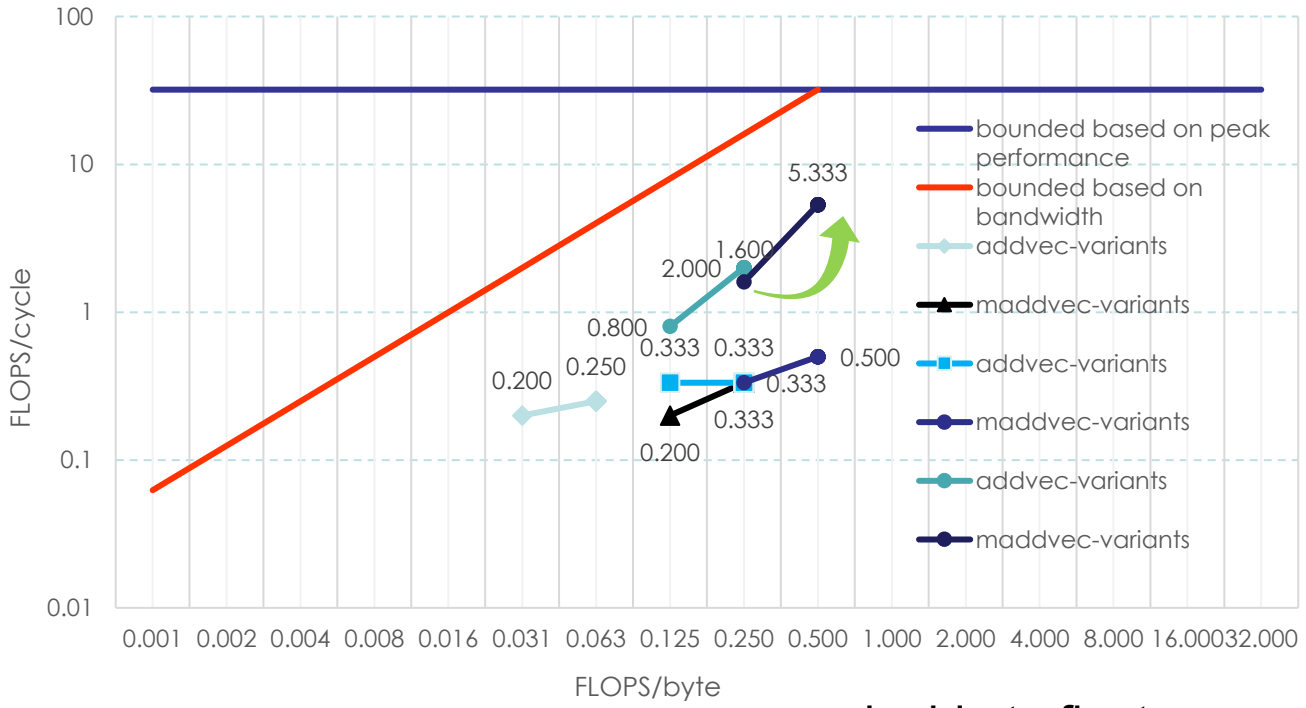


double to float

Ex.	Variants	float	double	w/ mac	w/ SWP	w/ loop vectoriz.	w/ SIMD	w/ FMA
Addvec	(1)		✓					
Addvec	(2)	✓						
Maddvec	(3)		✓	✓				
Maddvec	(4)	✓		✓				
Addvec	(5)		✓		✓			
Addvec	(6)	✓			✓			
Maddvec	(7)		✓	✓	✓			
Maddvec	(8)	✓		✓	✓			
Addvec	(9)		✓			✓	✓	
Addvec	(10)	✓				✓	✓	
Maddvec	(11)		✓			✓		✓
Maddvec	(12)	✓				✓		✓

Roofline model

- *Label each point in the Roofline model with the variant (1)…(12) of the table*



double to float

Ex.	Variants	float	double	w/ mac	w/ SWP	w/ loop vectoriz.	w/ SIMD	w/ FMA
Addvec	(1)		✓					
Addvec	(2)	✓						
Maddvec	(3)		✓	✓				
Maddvec	(4)	✓		✓				
Addvec	(5)		✓		✓			
Addvec	(6)	✓			✓			
Maddvec	(7)		✓	✓	✓			
Maddvec	(8)	✓		✓	✓			
Addvec	(9)		✓			✓	✓	
Addvec	(10)	✓				✓	✓	
Maddvec	(11)		✓			✓		✓
Maddvec	(12)	✓				✓		✓