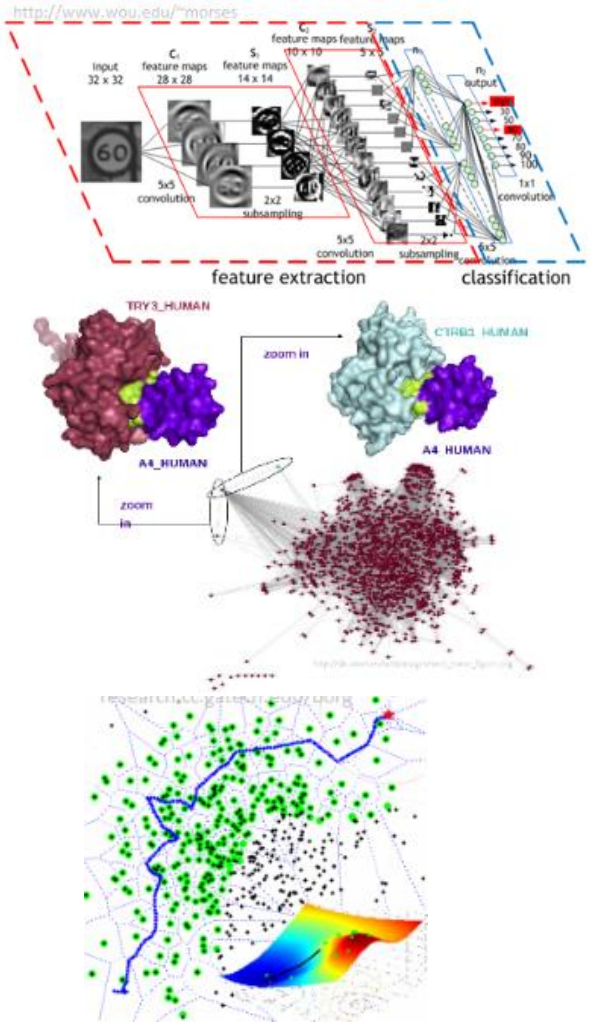
Flyte-MM: A Software Based Sub-Floating Point GEMM

Richard Veras (Louisiana State University)

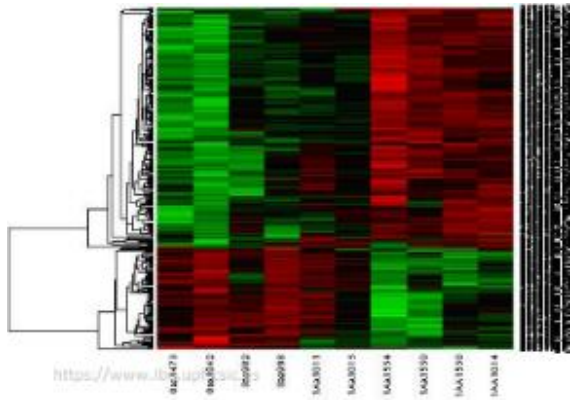
David Gregg (Trinity College, Dublin)

Want GEMM with High Dynamic Range but Low Precision

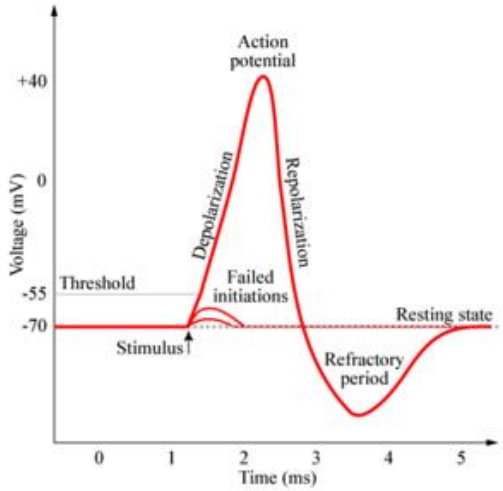
Targeting: Machine Learning and Analytics.

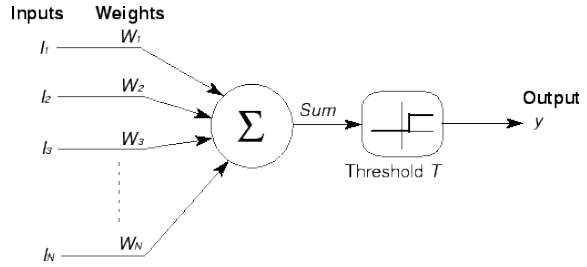


Have: Large Datasets and Constrained Systems.



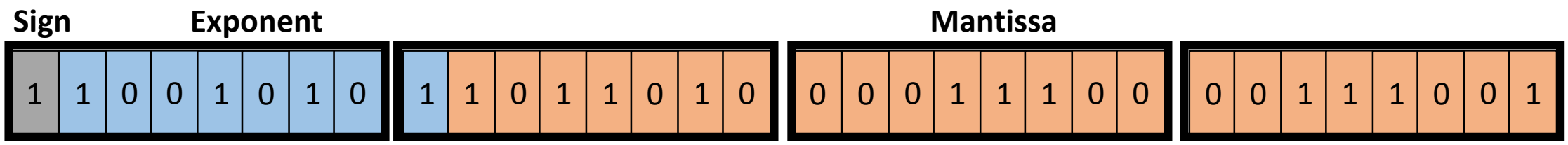
Need: High Dynamic Range, Not High Precision.



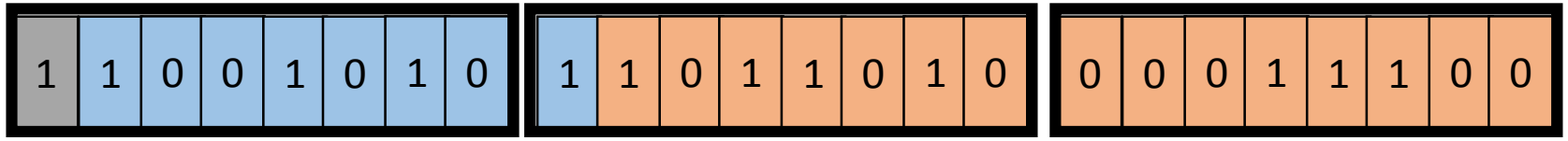


Flytes Preserve Dynamic Range at Reduced Space

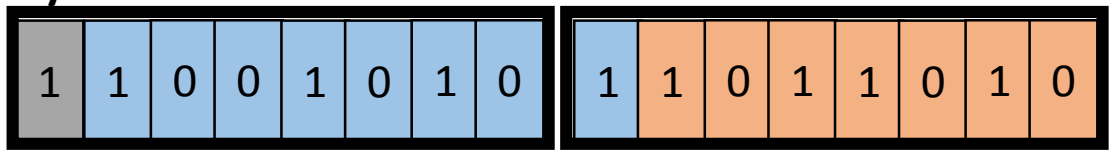
Float 32 (IEEE 754):



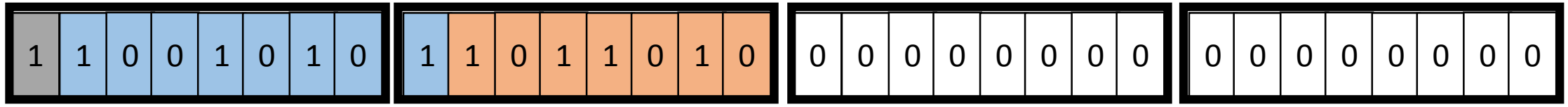
Flyte 24:



Flyte 16:



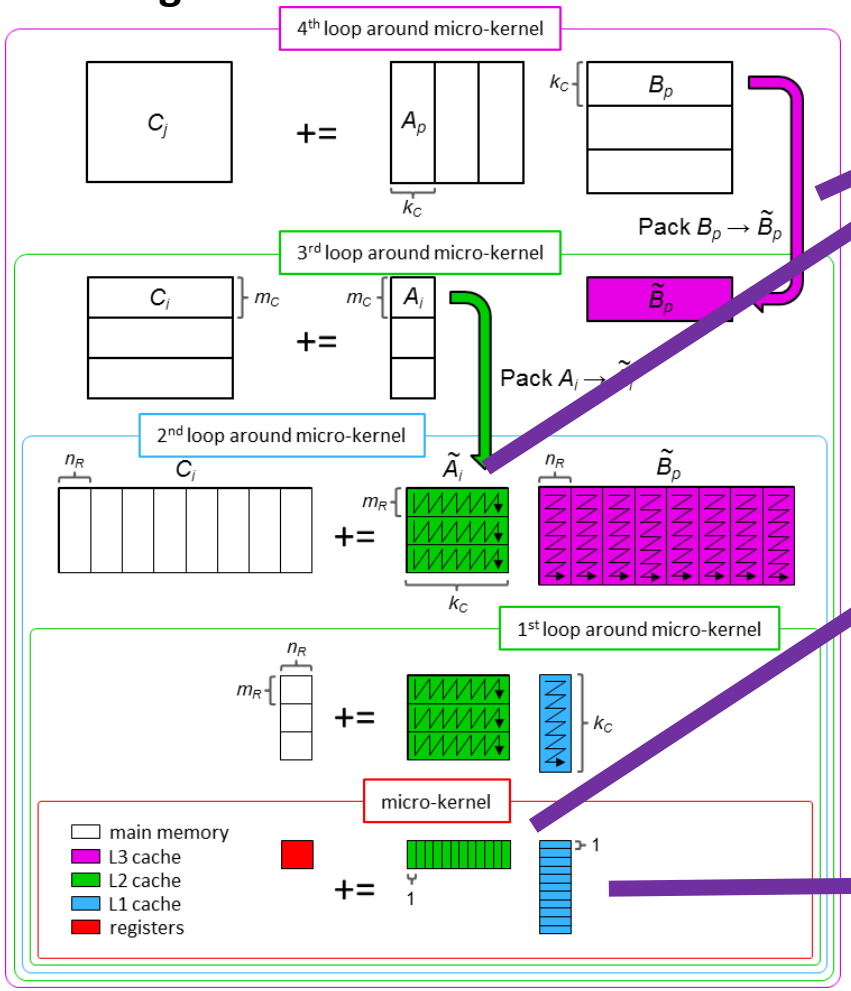
Flyte 16 to Float 32:



[Anderson, A., Muralidharan, S., Gregg, D. IEEE TOCS 2017]

Efficiently Converting Floats to Flyte in GEMM

BLIS Algorithm



Pack-b

Pack-a

Kern-a

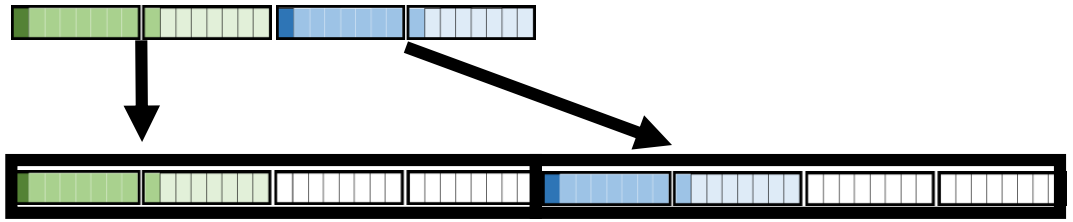
Kern-b

Convert in A/B in Pack: Fast, but cache block of float32.

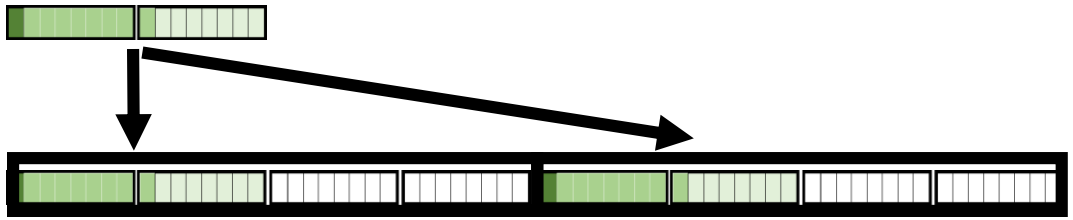
```
for( i = 0 .. mc )
  for( p = 0 .. kc )
      buffer_A[buff_addr(i,p)]
         = flYTE_to_flOAT(
             A[a_addr(i,p)] );
```

Convert in A in Kernel: Slow, but cache block of Flytes.

SIMD Load and Convert:

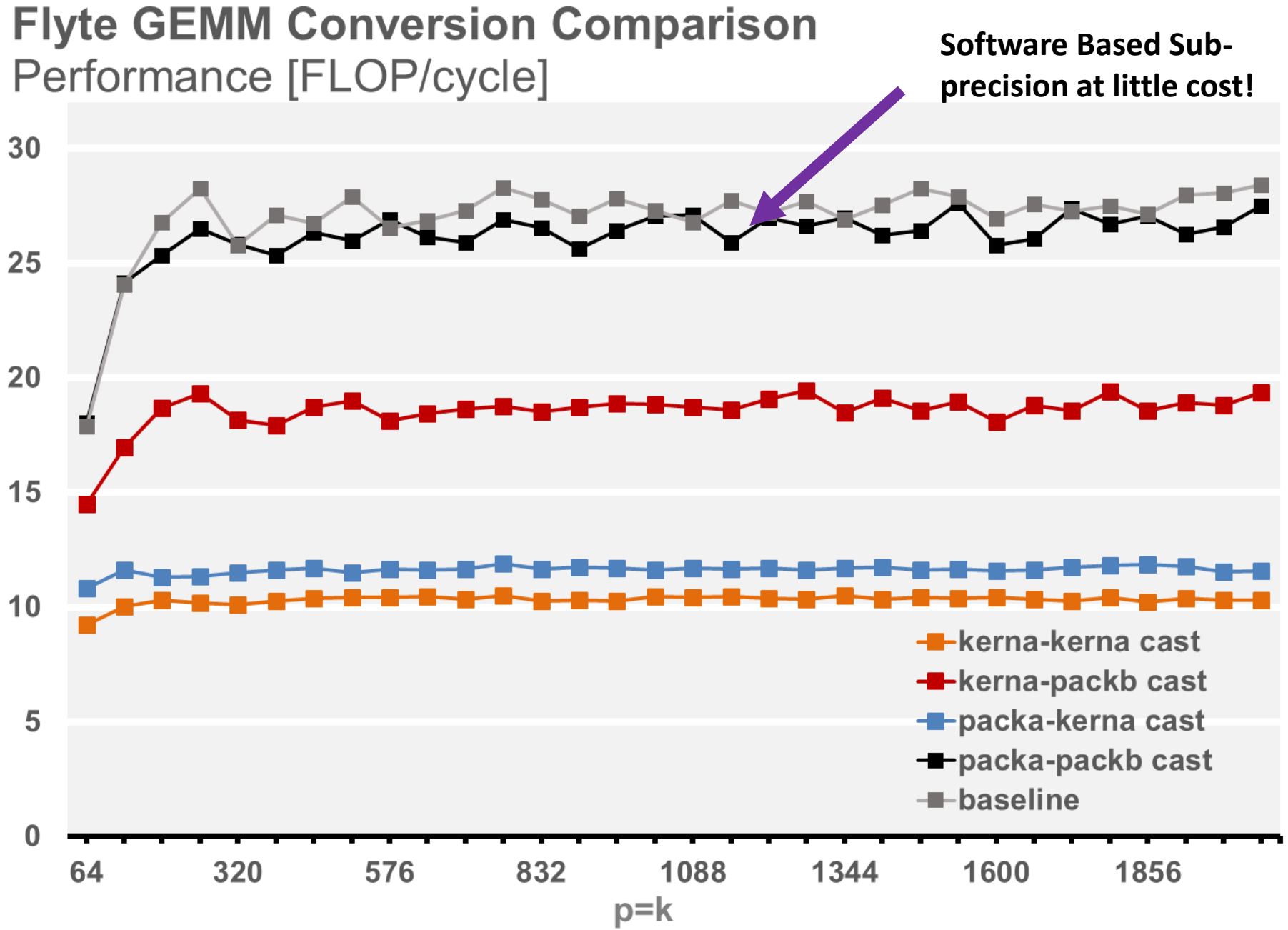


SIMD Broadcast and Convert:

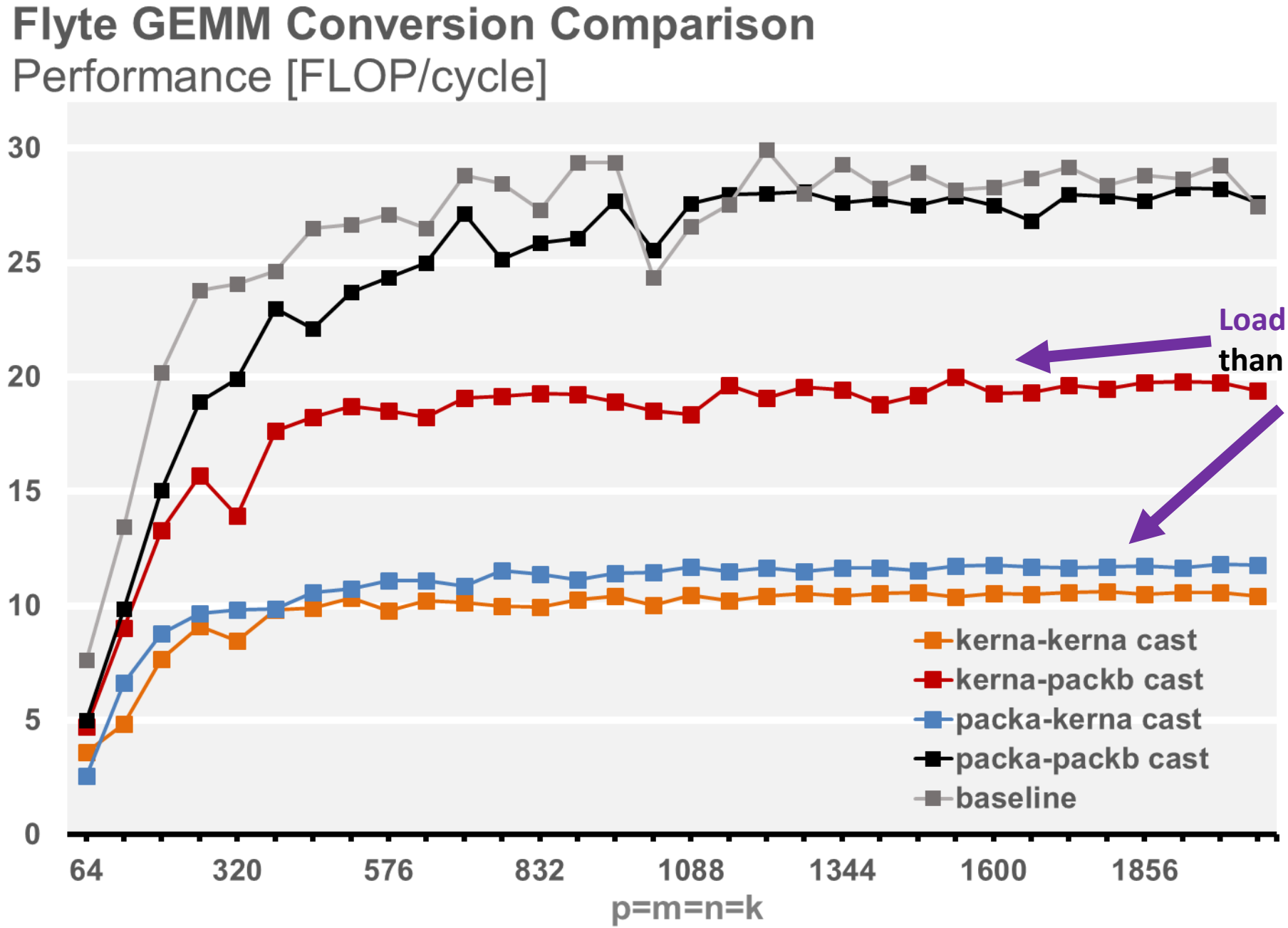


[van Zee, F., et. al. TOMS 2013]

A Family of Implementations



A Family of Implementations



Load Convert is cheaper than **Broadcast Convert**.

Exploring the parameters for each strategy.

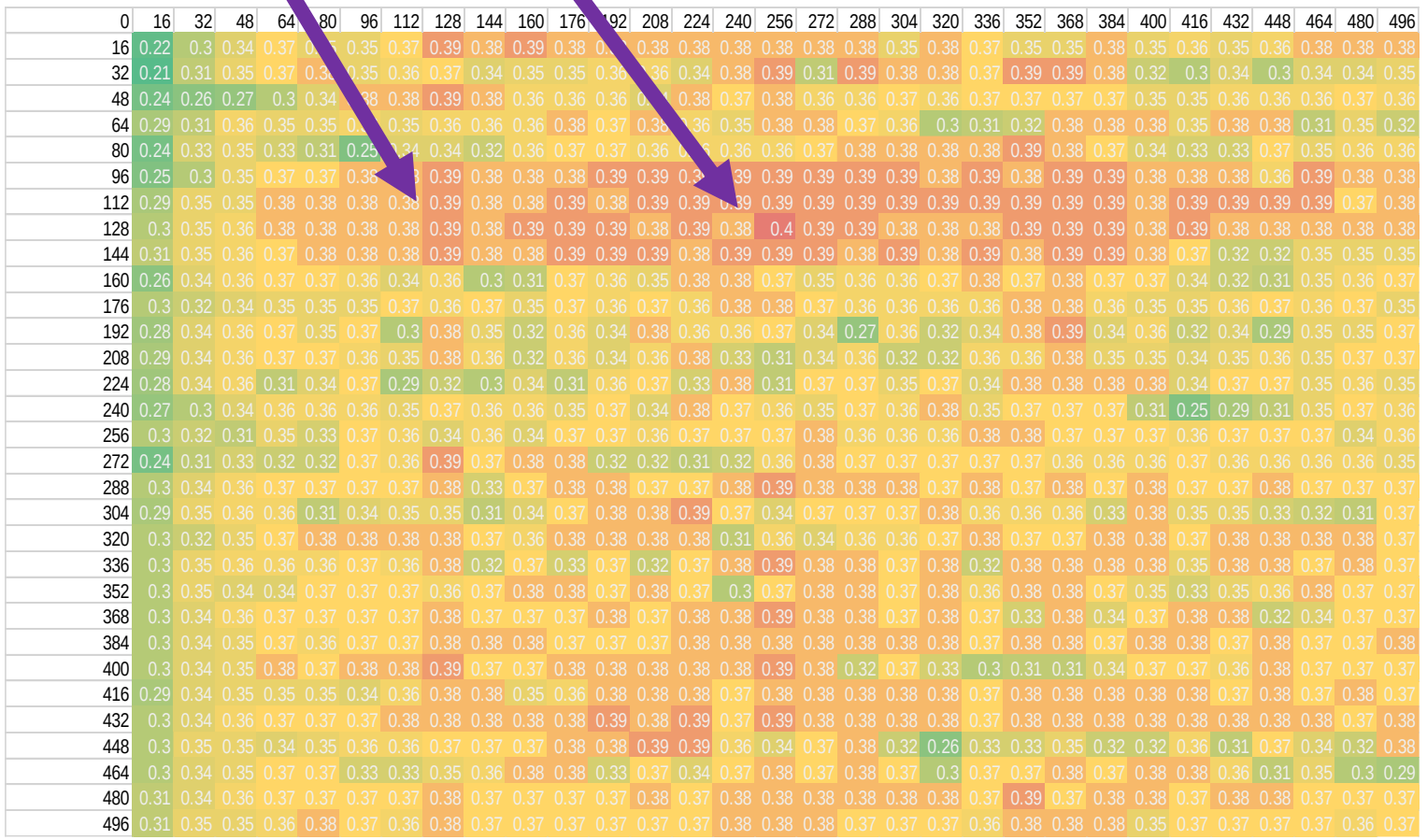
PackA-Pack B uses same block size as SGEMM, 6x16.

	1	2	3	4	5	6
8	0.1	0.15	0.3	0.4	0.54	0.15
16	0.12	0.17	0.58	0.79	0.93	1
24	0.14	0.18	0.64	0.97	0.54	0.56
32	0.08	0.14	0.69	0.48	0.55	0.37
40	0.08	0.14	0.66	0.55	0.48	0.32
48	0.08	0.11	0.56	0.42	0.47	0.4

PackA-KernB uses MrxNr 3x32.

	1	2	**3**	4	5	6
8	0.1	0.15	0.32	0.41	0.39	0.3
16	0.17	0.16	0.6	0.6	0.72	0.72
24	0.14	0.19	0.58	.77	0.75	0.57
32	0.11	0.17	1	0.79	0.47	0.37
40	0.1	0.17	0.7	0.57	0.48	0.32
48	0.09	0.19	0.81	0.52	0.43	0.57

Smaller data format for blocks allows for bigger block sizes (96x256[SGEMM] vs. 128x256[FLYTE24]).



Summary

- Flytes are **software based sub-precision** Floating Point.
  - Keep the sign and exponent and **truncate the mantissa**.
  - **Flyte** based **GEMM** desirable for **machine learning and analytics**.

- **Family of implementations** for computing on Flytes.
  - **BLIS provides** various **opportunities** to cast datatypes.
  - **Code Generation** makes **kernel** implementation **trivial**.

- Opportunity for **identifying optimal parameters**.
  - Kernel **throughput varies** with register **block dimensions**.
  - **Cache blocking** parameters **affected by datatype size**.