

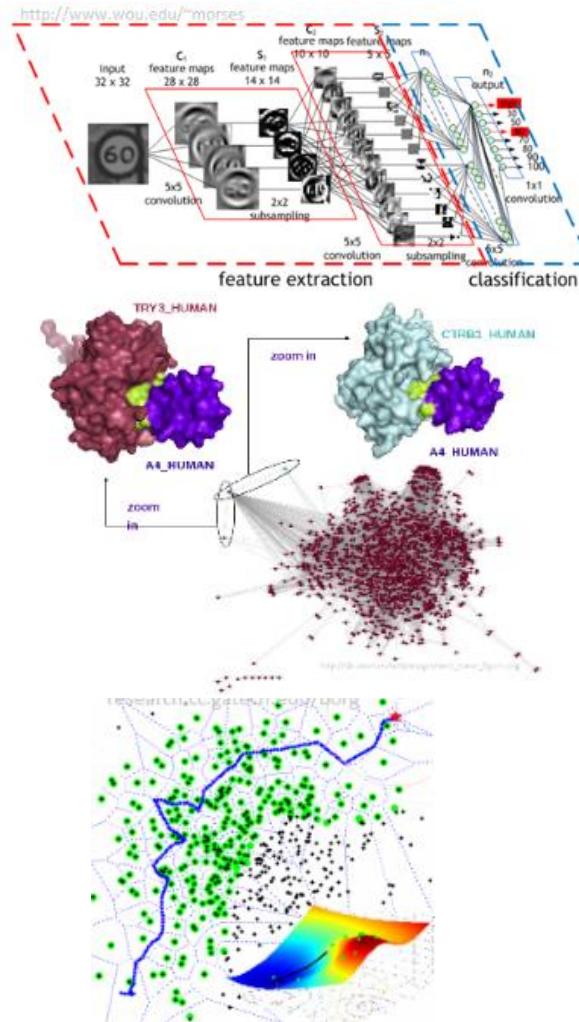
Flyte-MM: A Software Based Sub-Floating Point GEMM

Richard Veras (Louisiana State University)

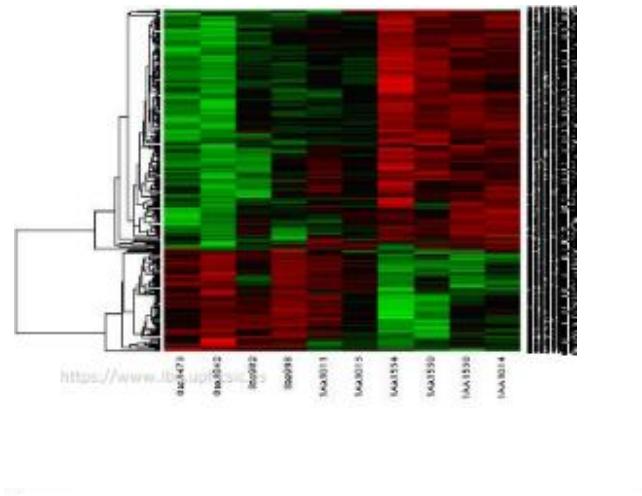
David Gregg (Trinity College, Dublin)

Want GEMM with High Dynamic Range but Low Precision

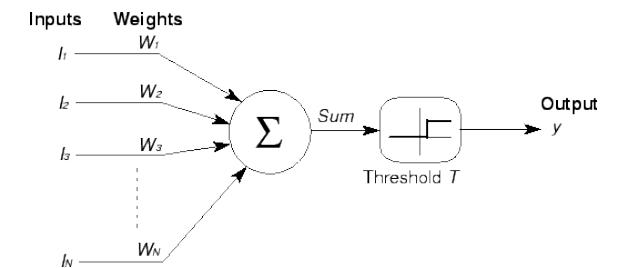
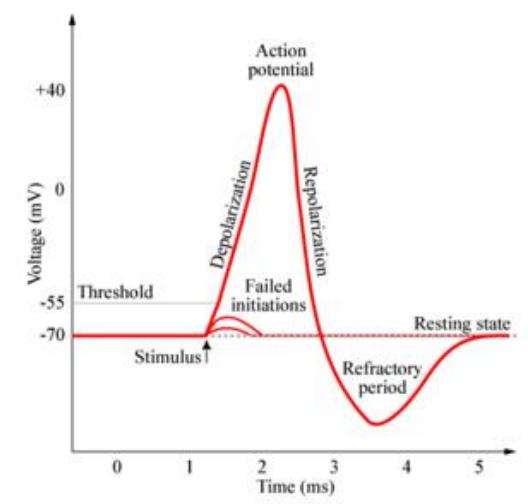
Targeting: Machine Learning and Analytics.



Have: Large Datasets and Constrained Systems.

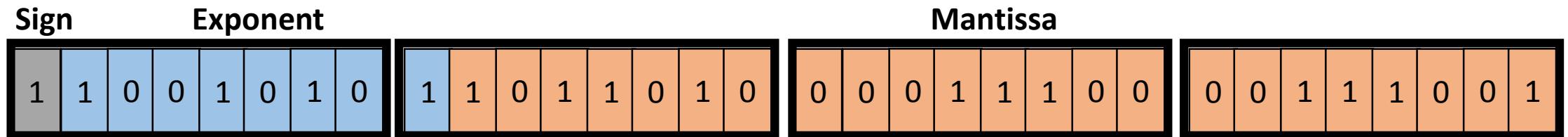


Need: High Dynamic Range, Not High Precision.

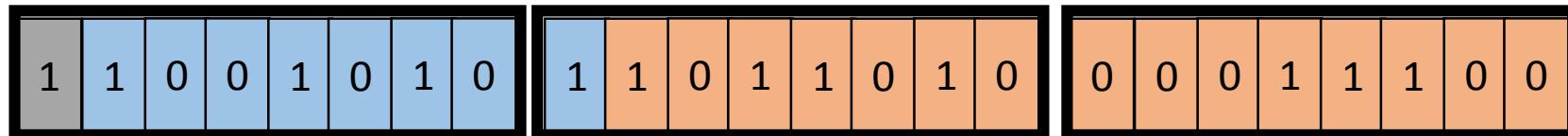


Flytes Preserve Dynamic Range at Reduced Space

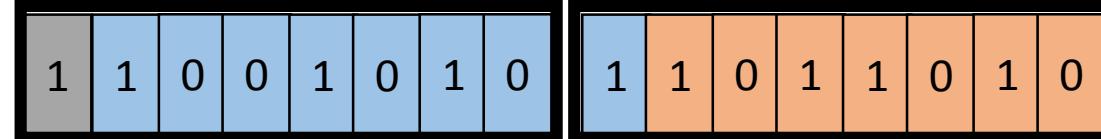
Float 32 (IEEE 754):



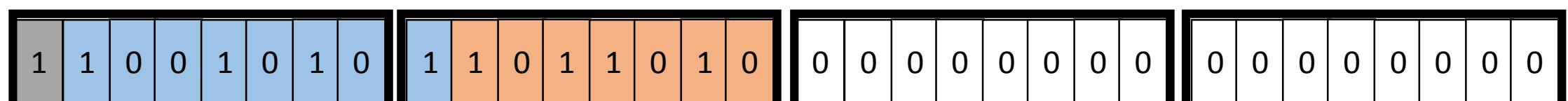
Flyte 24:



Flyte 16:

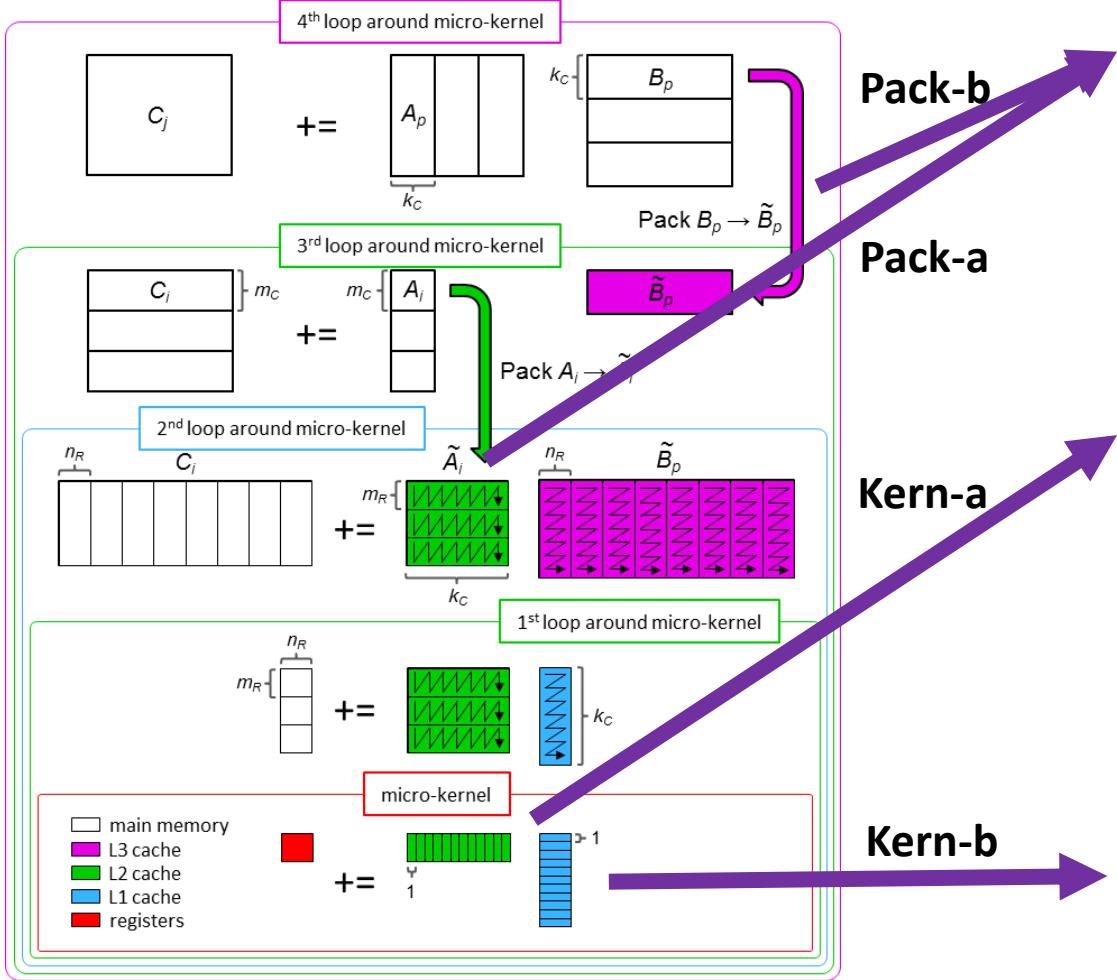


Flyte 16 to Float 32:



Efficiently Converting Floats to Flyte in GEMM

BLIS Algorithm

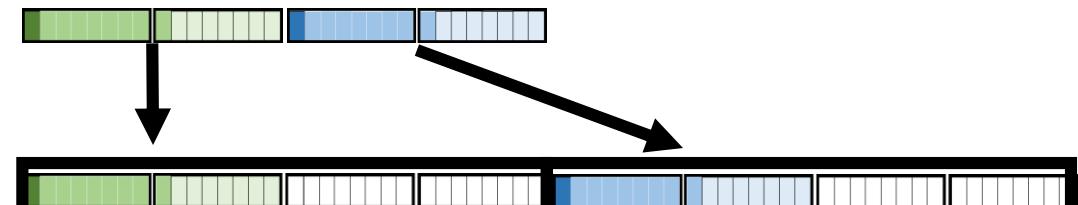


Convert in A/B in Pack: Fast, but cache block of float32.

```
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:

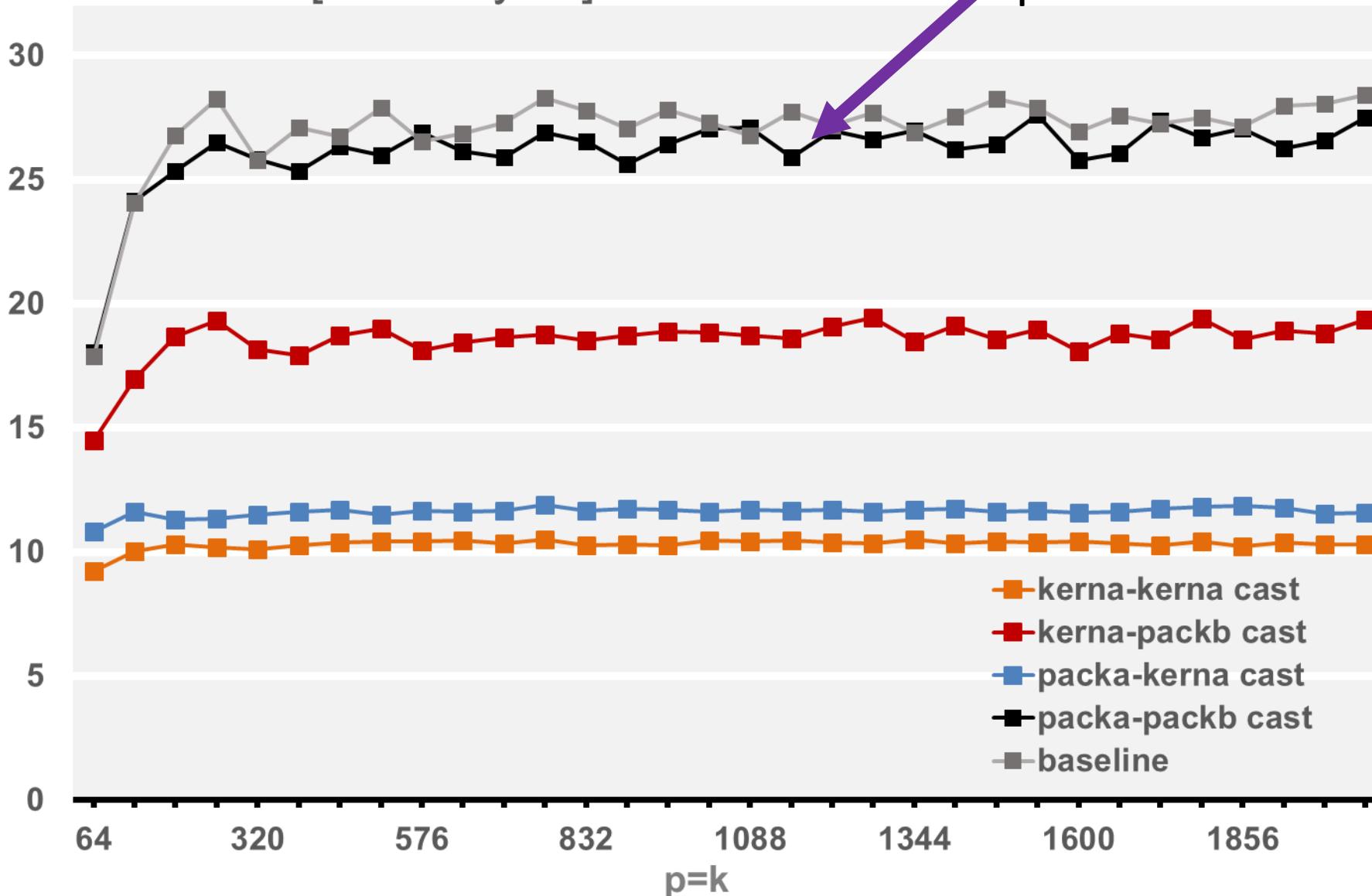


A Family of Implementations

Flyte GEMM Conversion Comparison

Performance [FLOP/cycle]

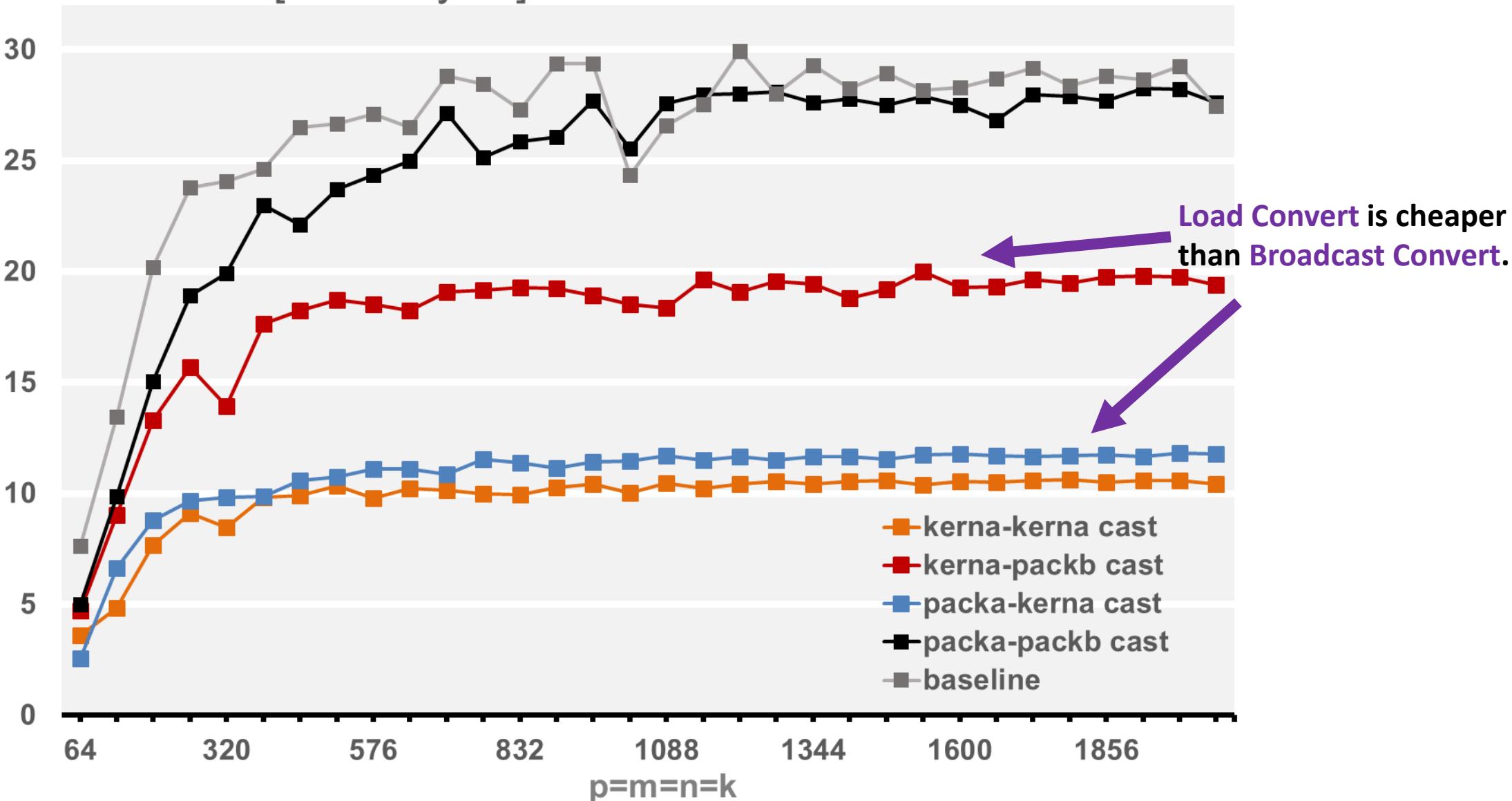
Software Based Sub-precision at little cost!



A Family of Implementations

Flyte GEMM Conversion Comparison

Performance [FLOP/cycle]



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	0.69	0.72	0.72
24	0.14	0.19	0.58	0.77	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]).

0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288	304	320	336	352	368	384	400	416	432	448	464	480	496				
16	0.22	0.3	0.34	0.37	0.35	0.35	0.37	0.39	0.38	0.39	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.35	0.38	0.37	0.35	0.35	0.38	0.35	0.36	0.35	0.36	0.38	0.38	0.38				
32	0.21	0.31	0.35	0.37	0.36	0.35	0.36	0.37	0.34	0.35	0.35	0.36	0.36	0.34	0.38	0.38	0.38	0.38	0.37	0.39	0.39	0.38	0.32	0.3	0.34	0.3	0.34	0.34	0.35	0.36	0.36				
48	0.24	0.26	0.27	0.3	0.34	0.38	0.38	0.39	0.38	0.36	0.36	0.36	0.36	0.38	0.37	0.38	0.36	0.36	0.37	0.37	0.37	0.37	0.35	0.35	0.36	0.36	0.37	0.36	0.36	0.36	0.36				
64	0.29	0.31	0.36	0.35	0.35	0.35	0.35	0.36	0.36	0.36	0.36	0.36	0.36	0.38	0.37	0.38	0.38	0.37	0.36	0.3	0.31	0.32	0.38	0.38	0.35	0.38	0.38	0.31	0.35	0.32					
80	0.24	0.33	0.35	0.33	0.31	0.25	0.34	0.34	0.32	0.36	0.37	0.37	0.36	0.36	0.36	0.36	0.37	0.38	0.38	0.38	0.39	0.37	0.34	0.33	0.33	0.37	0.35	0.36	0.36	0.36	0.36				
96	0.25	0.3	0.35	0.37	0.37	0.38	0.38	0.39	0.38	0.38	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.38	0.39	0.39	0.39	0.38	0.38	0.38	0.36	0.39	0.38	0.38	0.38	0.38				
112	0.29	0.35	0.35	0.38	0.38	0.38	0.38	0.39	0.38	0.38	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.37	0.38			
128	0.3	0.35	0.36	0.38	0.38	0.38	0.38	0.39	0.38	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38			
144	0.31	0.35	0.36	0.37	0.38	0.38	0.38	0.39	0.38	0.38	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.38	0.37	0.32	0.32	0.35	0.35	0.35	0.35			
160	0.26	0.34	0.36	0.37	0.37	0.36	0.34	0.36	0.3	0.31	0.37	0.36	0.35	0.38	0.37	0.35	0.36	0.36	0.37	0.38	0.37	0.37	0.37	0.34	0.32	0.31	0.35	0.36	0.37	0.37	0.37	0.35			
176	0.3	0.32	0.34	0.35	0.35	0.35	0.37	0.36	0.37	0.35	0.37	0.36	0.37	0.36	0.38	0.38	0.37	0.36	0.36	0.36	0.38	0.38	0.36	0.35	0.35	0.36	0.37	0.36	0.37	0.35	0.35				
192	0.28	0.34	0.36	0.37	0.35	0.37	0.3	0.38	0.35	0.32	0.36	0.34	0.38	0.36	0.36	0.37	0.34	0.27	0.36	0.32	0.34	0.38	0.39	0.34	0.36	0.32	0.34	0.35	0.35	0.37	0.37	0.37			
208	0.29	0.34	0.36	0.37	0.37	0.36	0.35	0.38	0.36	0.32	0.36	0.34	0.36	0.38	0.33	0.31	0.34	0.36	0.32	0.32	0.36	0.38	0.35	0.35	0.34	0.35	0.36	0.37	0.37	0.37	0.37				
224	0.28	0.34	0.36	0.31	0.34	0.37	0.29	0.32	0.3	0.34	0.31	0.36	0.37	0.33	0.38	0.31	0.37	0.37	0.35	0.37	0.34	0.38	0.38	0.38	0.34	0.37	0.37	0.35	0.36	0.35	0.36	0.35			
240	0.27	0.3	0.34	0.36	0.36	0.36	0.35	0.37	0.36	0.36	0.35	0.37	0.36	0.35	0.37	0.34	0.38	0.37	0.36	0.35	0.37	0.37	0.37	0.31	0.25	0.29	0.31	0.35	0.37	0.36	0.36	0.36			
256	0.3	0.32	0.31	0.35	0.33	0.37	0.36	0.34	0.36	0.34	0.37	0.37	0.36	0.37	0.37	0.37	0.38	0.36	0.36	0.38	0.38	0.37	0.37	0.37	0.36	0.37	0.37	0.37	0.34	0.36	0.36	0.36	0.35		
272	0.24	0.31	0.33	0.32	0.32	0.37	0.36	0.39	0.37	0.38	0.38	0.32	0.32	0.31	0.32	0.36	0.38	0.37	0.37	0.37	0.37	0.36	0.36	0.36	0.37	0.36	0.36	0.36	0.36	0.36	0.36	0.35			
288	0.3	0.34	0.36	0.37	0.37	0.37	0.37	0.38	0.33	0.37	0.38	0.38	0.37	0.37	0.38	0.38	0.38	0.37	0.38	0.37	0.38	0.37	0.38	0.37	0.38	0.37	0.37	0.38	0.37	0.37	0.37	0.37	0.37		
304	0.29	0.35	0.36	0.36	0.31	0.34	0.35	0.35	0.31	0.34	0.37	0.38	0.38	0.39	0.37	0.34	0.37	0.37	0.37	0.38	0.36	0.36	0.36	0.38	0.37	0.37	0.37	0.36	0.36	0.36	0.35	0.33	0.31	0.37	
320	0.3	0.32	0.35	0.37	0.38	0.38	0.38	0.37	0.36	0.38	0.38	0.38	0.38	0.38	0.31	0.36	0.34	0.36	0.36	0.37	0.38	0.37	0.37	0.38	0.38	0.37	0.38	0.38	0.38	0.38	0.37	0.37	0.37	0.37	
336	0.3	0.35	0.36	0.36	0.37	0.36	0.38	0.32	0.37	0.33	0.37	0.32	0.37	0.38	0.39	0.38	0.37	0.38	0.32	0.38	0.38	0.35	0.38	0.37	0.38	0.37	0.35	0.33	0.35	0.36	0.38	0.37	0.37	0.37	
352	0.3	0.35	0.34	0.34	0.37	0.37	0.36	0.37	0.38	0.38	0.37	0.38	0.37	0.3	0.37	0.38	0.38	0.37	0.38	0.36	0.38	0.38	0.37	0.35	0.33	0.35	0.36	0.38	0.37	0.37	0.37	0.37	0.37	0.37	
368	0.3	0.34	0.36	0.37	0.37	0.37	0.38	0.37	0.37	0.38	0.37	0.38	0.38	0.39	0.38	0.38	0.38	0.38	0.37	0.33	0.38	0.34	0.37	0.38	0.38	0.32	0.34	0.37	0.38	0.32	0.34	0.37	0.37	0.37	0.37
384	0.3	0.34	0.35	0.37	0.36	0.37	0.37	0.38	0.38	0.38	0.37	0.37	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.37	0.38	0.38	0.37	0.38	0.37	0.38	0.38	0.37	0.38	0.37	0.38	0.37	0.38	0.37	0.38
400	0.3	0.34	0.35	0.38	0.37	0.38	0.38	0.39	0.37	0.37	0.38	0.38	0.38	0.38	0.39	0.38	0.38	0.38	0.32	0.37	0.33	0.3	0.31	0.31	0.34	0.37	0.37	0.36	0.38	0.37	0.37	0.37	0.37	0.37	
416	0.29	0.34	0.35	0.35	0.34	0.36	0.38	0.38	0.35	0.36	0.38	0.38	0.38	0.37	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.37	0.38	0.37
432	0.3	0.34	0.36	0.37	0.37	0.38	0.38	0.38	0.38	0.38	0.39	0.37	0.39	0.38	0.38	0.38	0.38	0.38	0.37	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38
448	0.3	0.35	0.34	0.35	0.36	0.36	0.37	0.37	0.38	0.38	0.39	0.39	0.36	0.34	0.37	0.38	0.38	0.38	0.38	0.37	0.38	0.32	0.26	0.33	0.33	0.35	0.32	0.32	0.36	0.31	0.37	0.34	0.32	0.38	
464	0.3	0.34	0.35	0.37	0.37	0.33	0.33	0.35	0.36	0.38	0.38	0.33	0.37	0.34	0.37	0.38	0.37	0.38	0.37	0.3	0.37	0.37	0.38	0.37	0.38	0.36	0.31	0.35							

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**.