NHR FAU

FAU
Friedrich-Alexander-Universität
Erlangen-Nürnberg

MICROARCHITECTURAL COMPARISON AND IN-CORE MODELING OF STATE-OF-THE-ART CPUS: GRACE, SAPPHIRE RAPIDS, AND GENOA

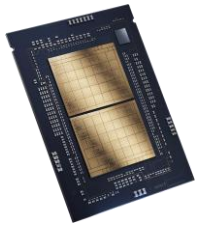
Jan Laukemann, Georg Hager, Gerhard Wellein

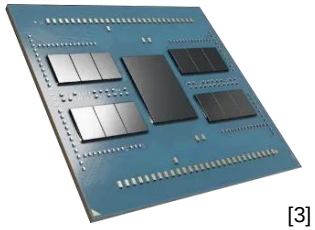
Erlangen National High Performance Computing Center (NHR@FAU)
Friedrich-Alexander University Erlangen-Nürnberg

SC24
Atlanta, GA | hpc creates.

A new player in the (CPU) game


[1]

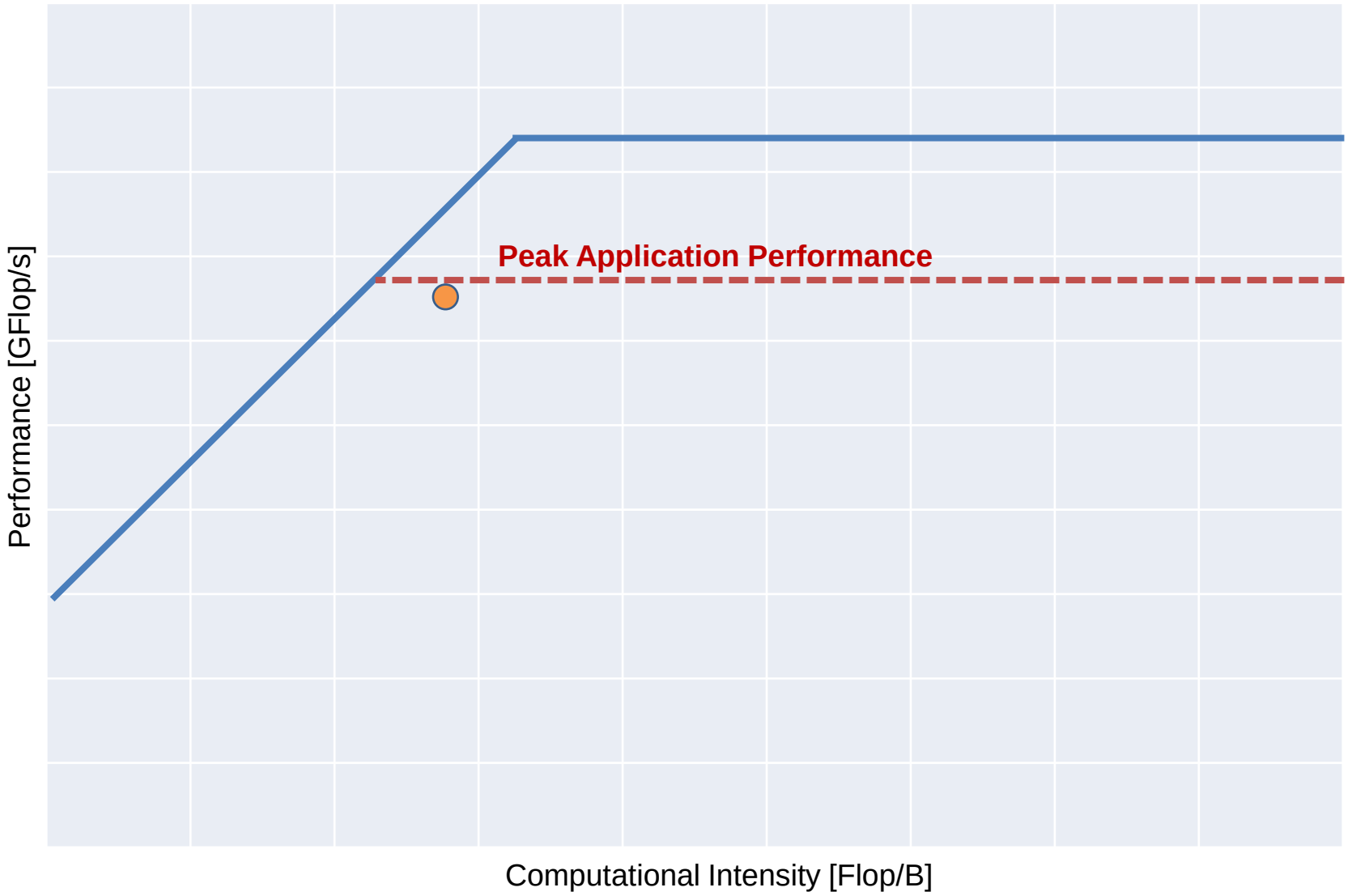

[2]


[3]

	Grace CPU Superchip (GCS)	Sapphire Rapids (SPR)	Genoa
cores	72	52	96
frequency (base/max)	3.4 GHz	2.0 – 3.8 GHz	2.55 – 3.7 GHz
Double-precision peak (meas.)	3.82 Tflop/s	3.49 Tflop/s	5.1 Tflop/s
TDP	250 W	350 W	400 W
Power efficiency	15.28 GFLOPS/W	9.97 GFLOPS/W	12.75 GFLOPS/W
Max mem BW (meas.)	467 GB/s	273 GB/s	375 GB/s

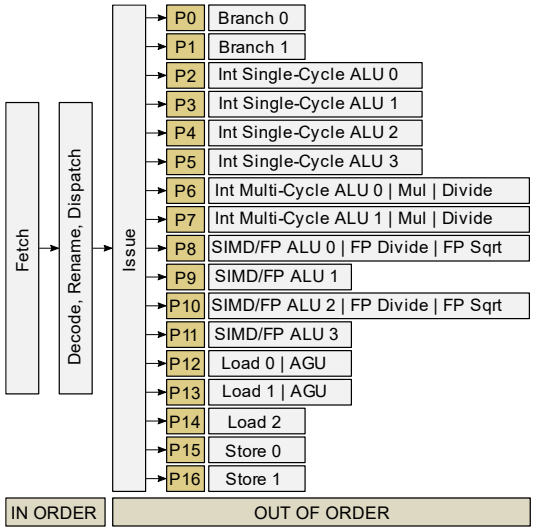
How do the cores actually perform?

Building an in-core performance model



Building an in-core performance model

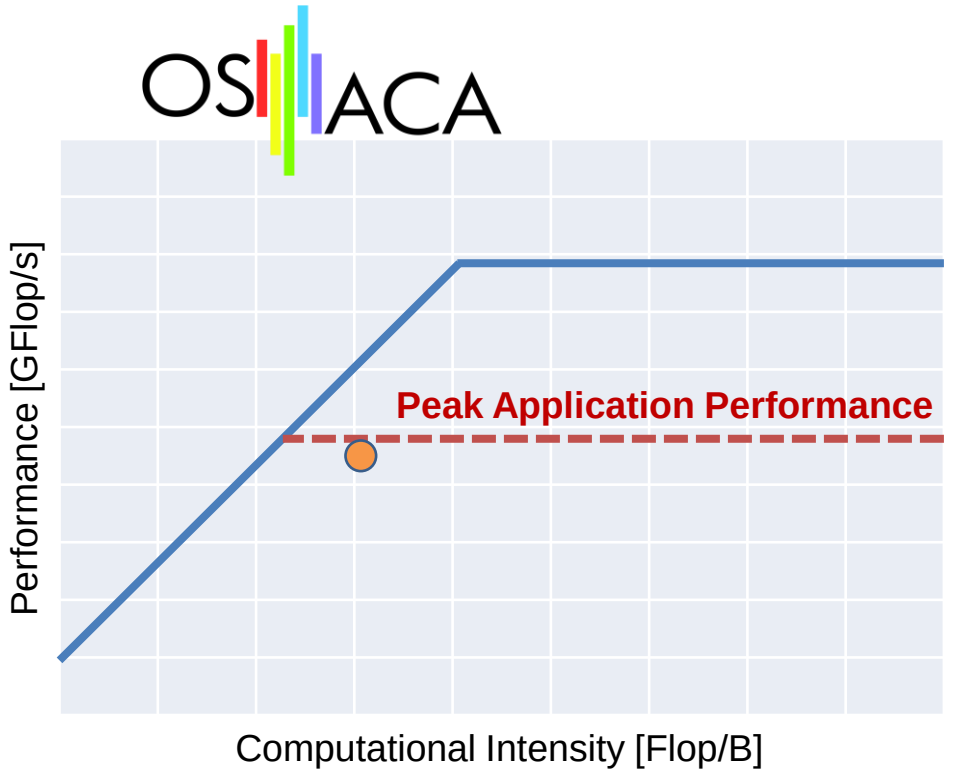
- port model
  - abstraction of superscalar and OoO abilities



- ASM benchmarks
  - gather performance for each instr

```
#define INSTR fmla
#define NINST 6
#define N x0

.arch armv8.2-a+sve
.text

mov     x4, N
ptrue   p0.d
fcpy    z0.d, p0/m, #1.000

loop:
    subs      x4, x4, #1
    INSTR     z1.d, p0/m, z0.d, z0.d
    INSTR     z2.d, p0/m, z0.d, z0.d
    INSTR     z3.d, p0/m, z0.d, z0.d
    INSTR     z4.d, p0/m, z0.d, z0.d
    INSTR     z5.d, p0/m, z0.d, z0.d
    INSTR     z6.d, p0/m, z0.d, z0.d
    bne       loop
done:

    ret
```

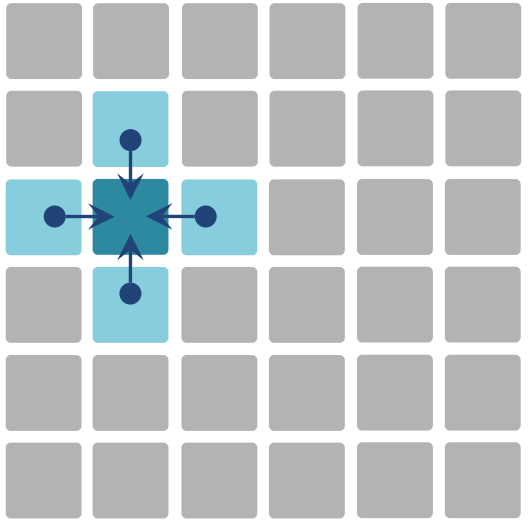
- Performance model
  - throughput
  - latency & dependencies
  - port occupation



Building an in-core performance model

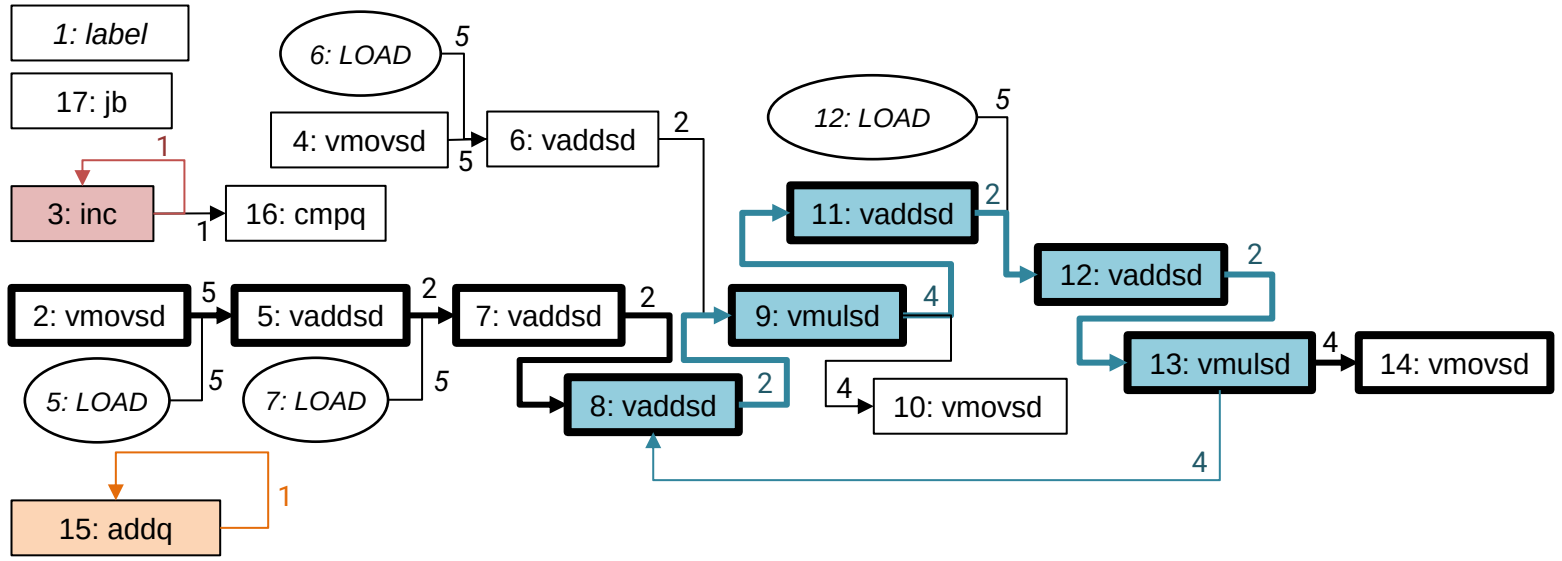
- 2D Gauss-Seidel method

```
for (int i=1; i<NI-1; ++i)
  for (int k=1; k<NK-1; ++k)
    phi[i][k] = 0.25 * (
        phi[i][k-1] + phi[i+1][k] +
        phi[i][k+1] + phi[i-1][k]
    );
```



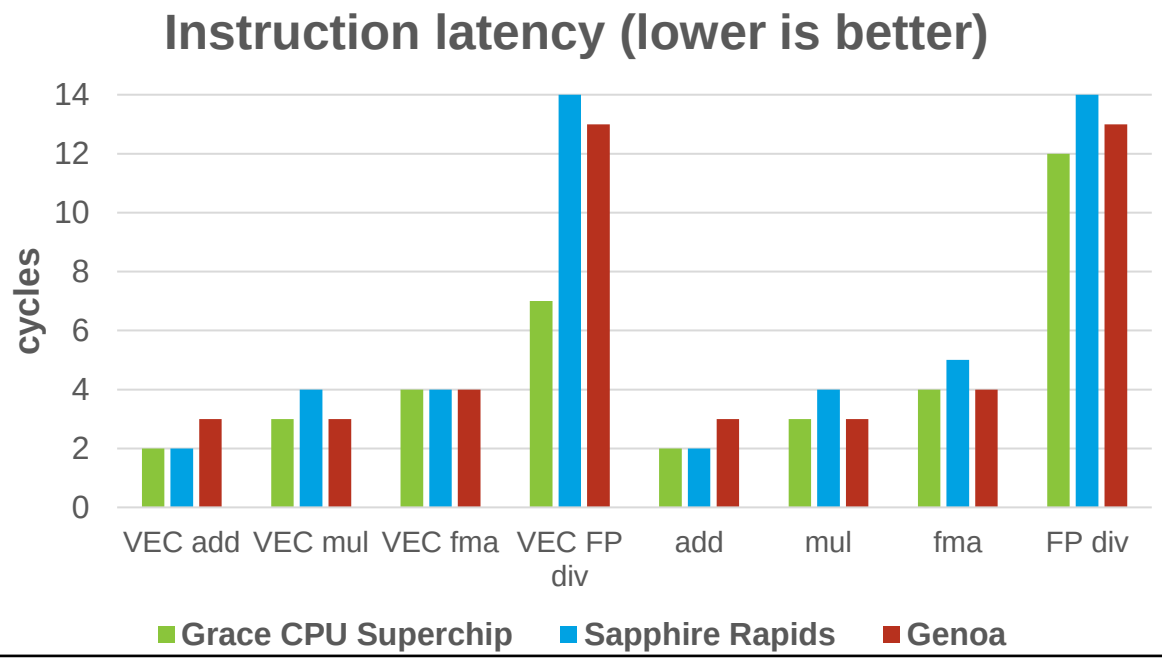
```
 1  ..B1.72:
 2  vmovsd    8(%r10,%r11), %xmm2
 3  incq      %rdx
 4  vmovsd    16(%r10,%r11), %xmm5
 5  vaddsd    16(%r10,%rsi), %xmm2, %xmm3
 6  vaddsd    24(%r10,%rsi), %xmm5, %xmm7
 7  vaddsd    8(%r10,%r13), %xmm3, %xmm4
 8  vaddsd    %xmm1, %xmm4, %xmm1
 9  vmulsd    %xmm0, %xmm1, %xmm6
10  vmovsd    %xmm6, 8(%r10,%rsi)
11  vaddsd    %xmm7, %xmm6, %xmm8
12  vaddsd    16(%r10,%r13), %xmm8, %xmm9
13  vmulsd    %xmm0, %xmm9, %xmm1
14  vmovsd    %xmm1, 16(%r10,%rsi)
15  addq      $16, %r10
16  cmpq      %r15, %rdx
17  jb        ..B1.72
```

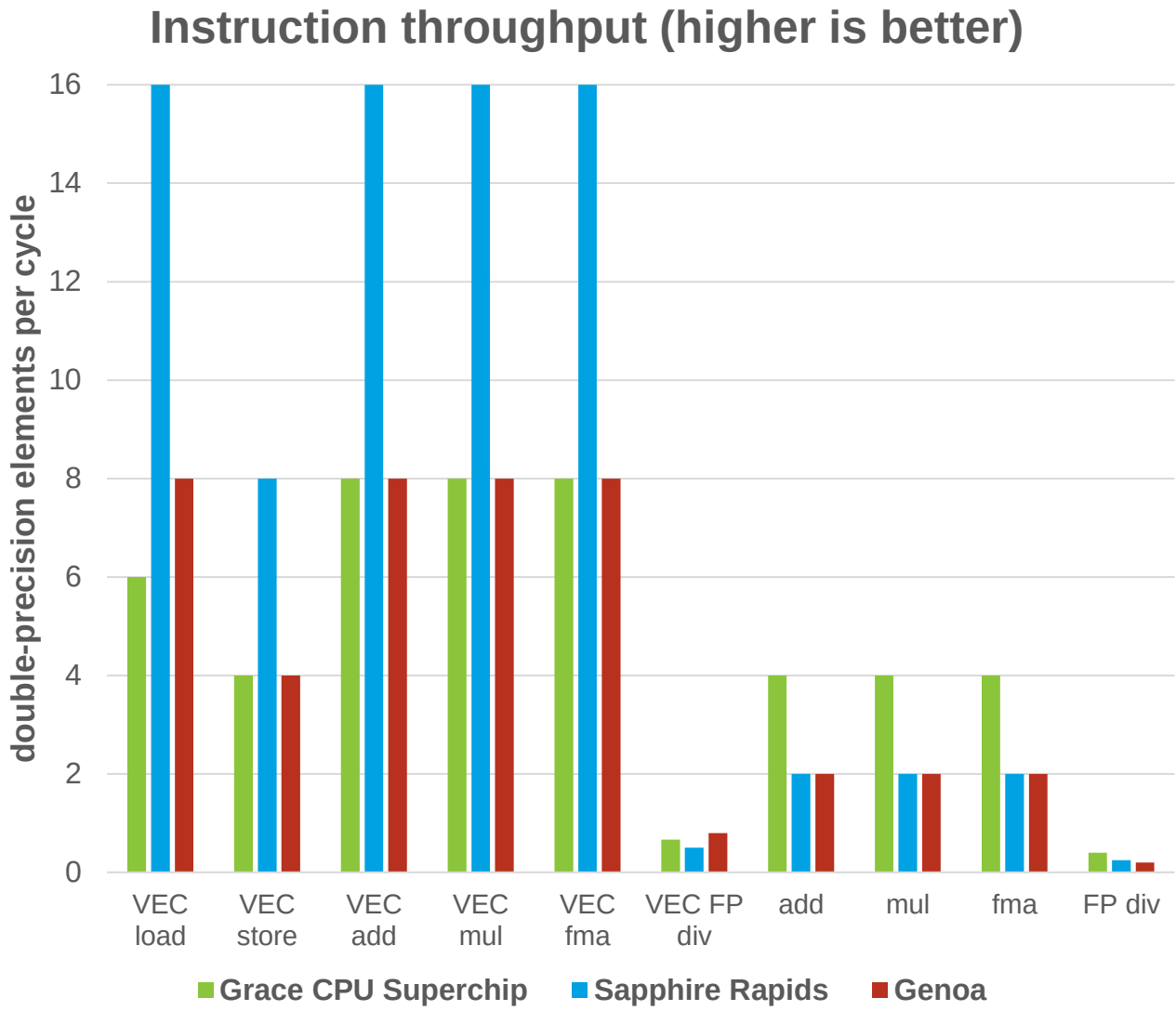
OSACA



In-core properties

	GCS	SPR	Genoa
#ports	17	12	13
SIMD-width	16 B	64 B	32 B
INT units	6	5	4
FP/vector units	4	3	4





Model validation

- microbenchmarks

1D STREAMING

copy	`a[] = b[]`
add	`a[] = b[] + c[]`
update	`a[] = s * a[]`
store	`a[] = s`
sum reduction	`s = s + a[]`
DAXPY	`a[] = a[] + s * b[]`
STREAM Triad	`a[] = b[] + s * c[]`
Schönauer Triad	`a[] = b[] + c[] * d[]`

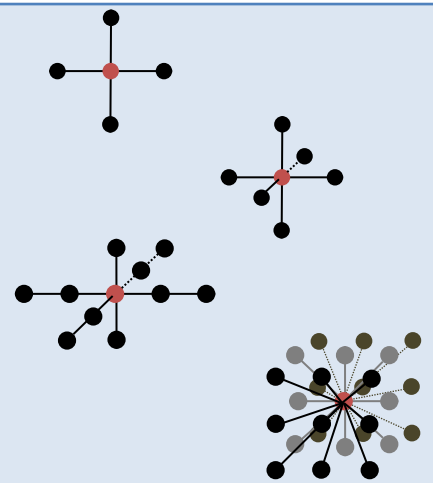
2D/3D STENCILS

Jacobi 2D-5pt

Jacobi 3D-7pt

Jacobi 3D-r3-11pt

Jacobi 3D-27pt



LATENCY-BOUND

Gauss-Seidel

```
a[j][i] = s * (
            a[j+1][i] +
        a[j][i-1] + a[j][i+1] +
            a[j-1][i]
            )
```

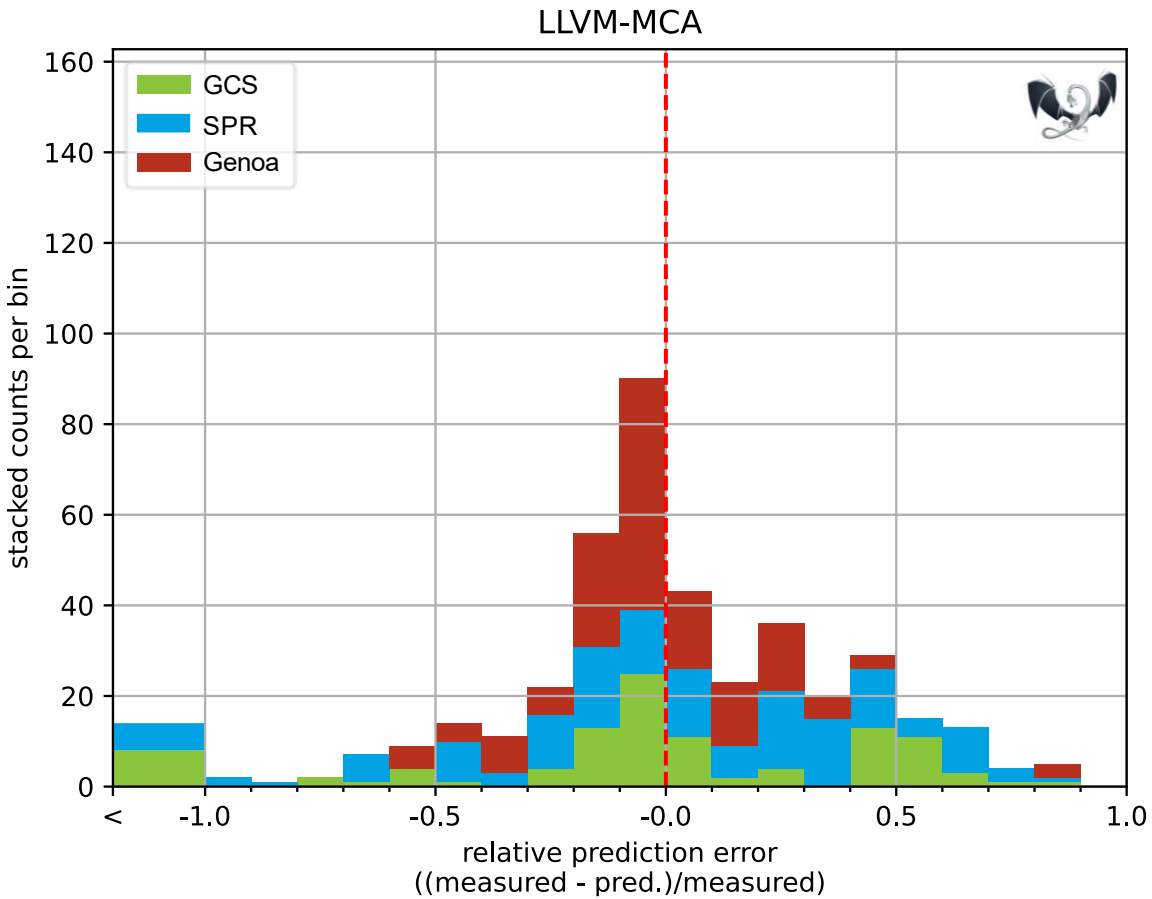
π by integration $\int_0^1 \frac{4}{1+x^2} dx$

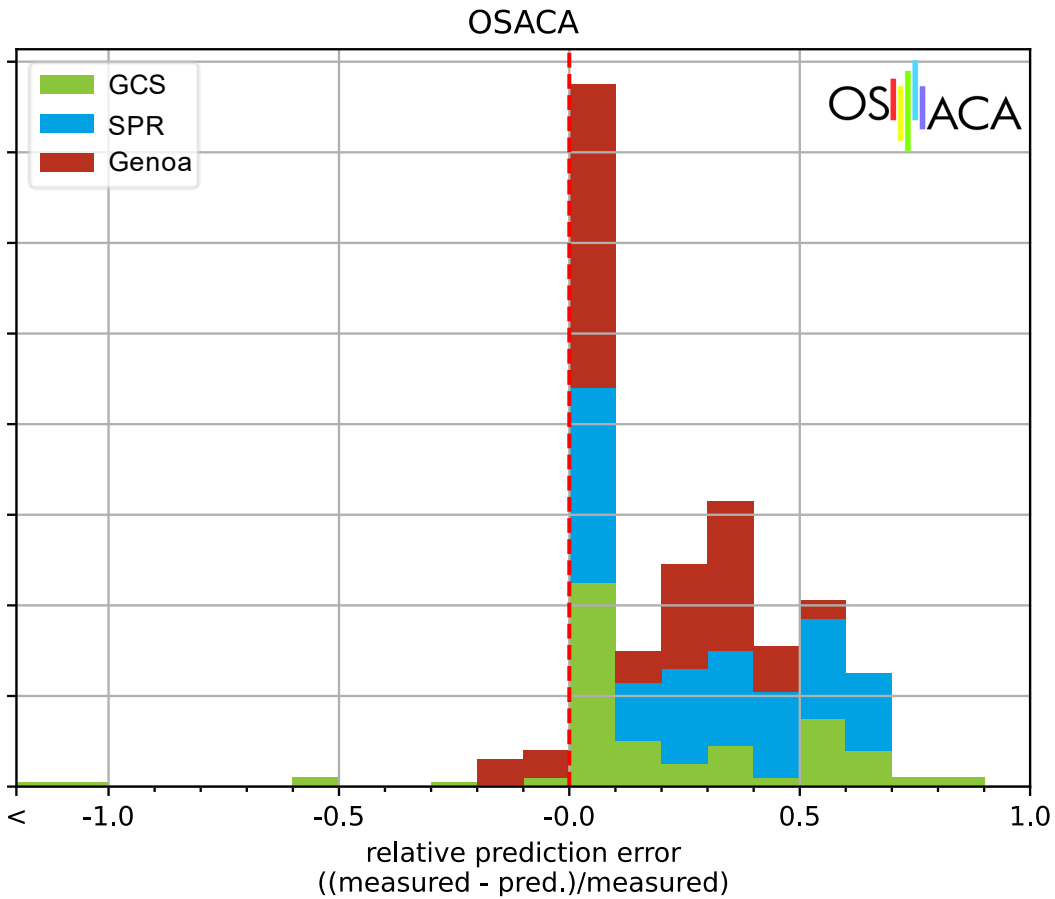
Model validation

≥ 2x faster
than prediction

over-prediction

under-prediction



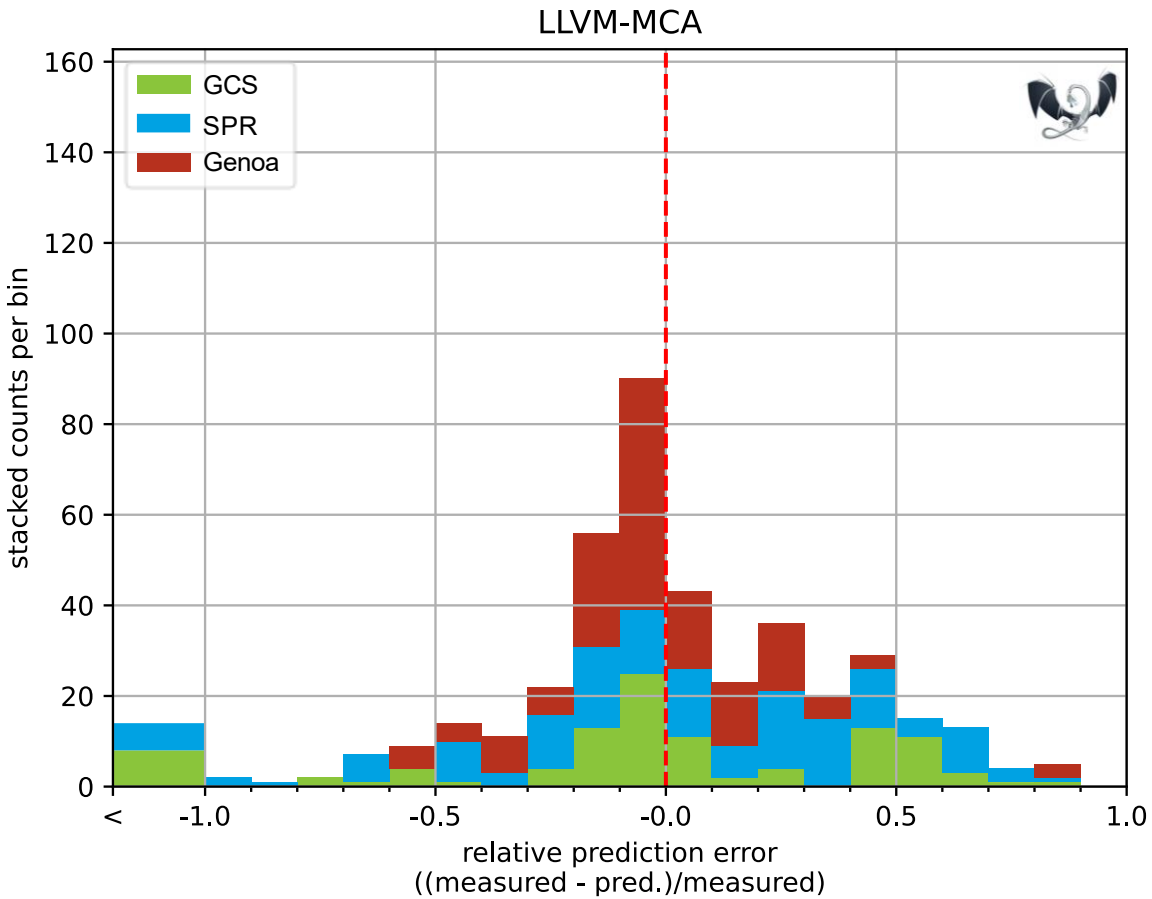


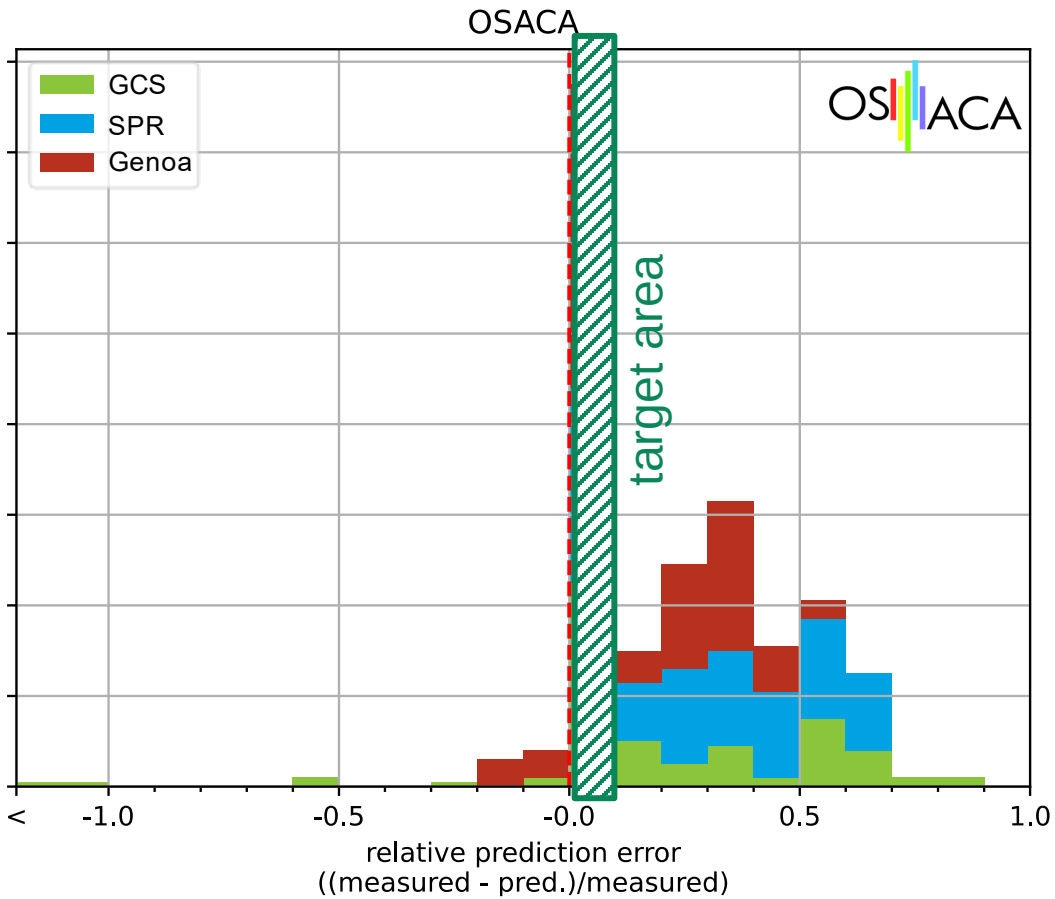
Model validation

≥ 2x faster
than prediction

over-prediction

under-prediction

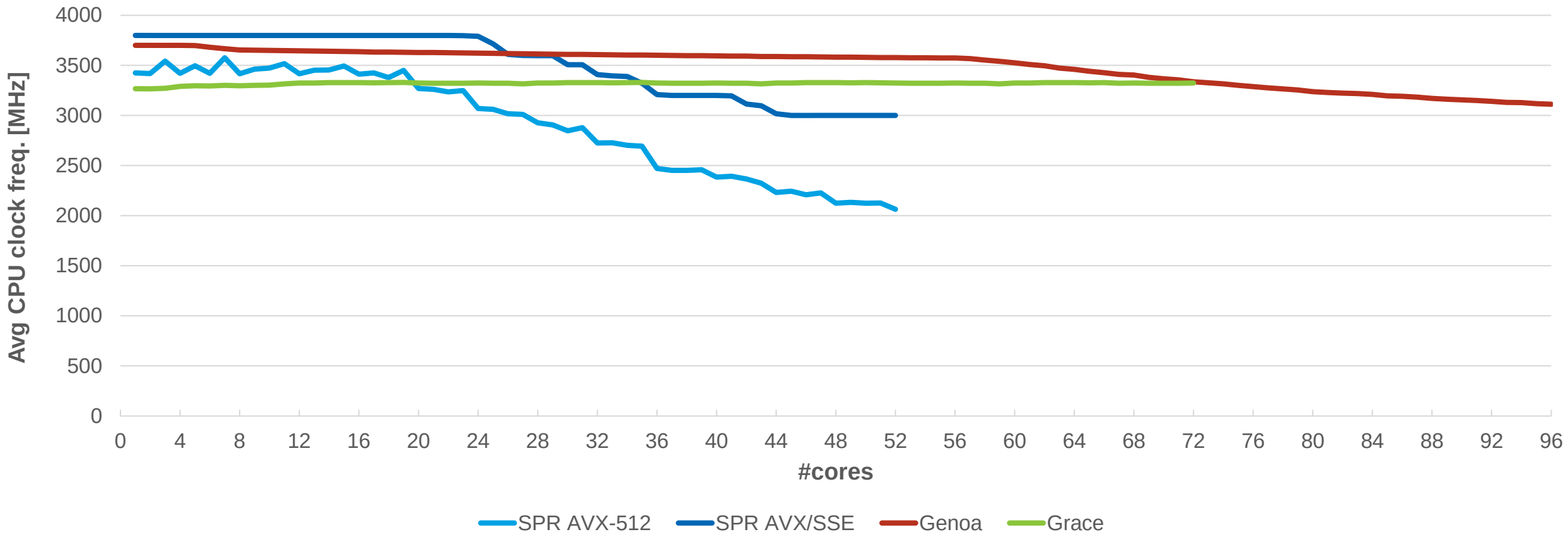




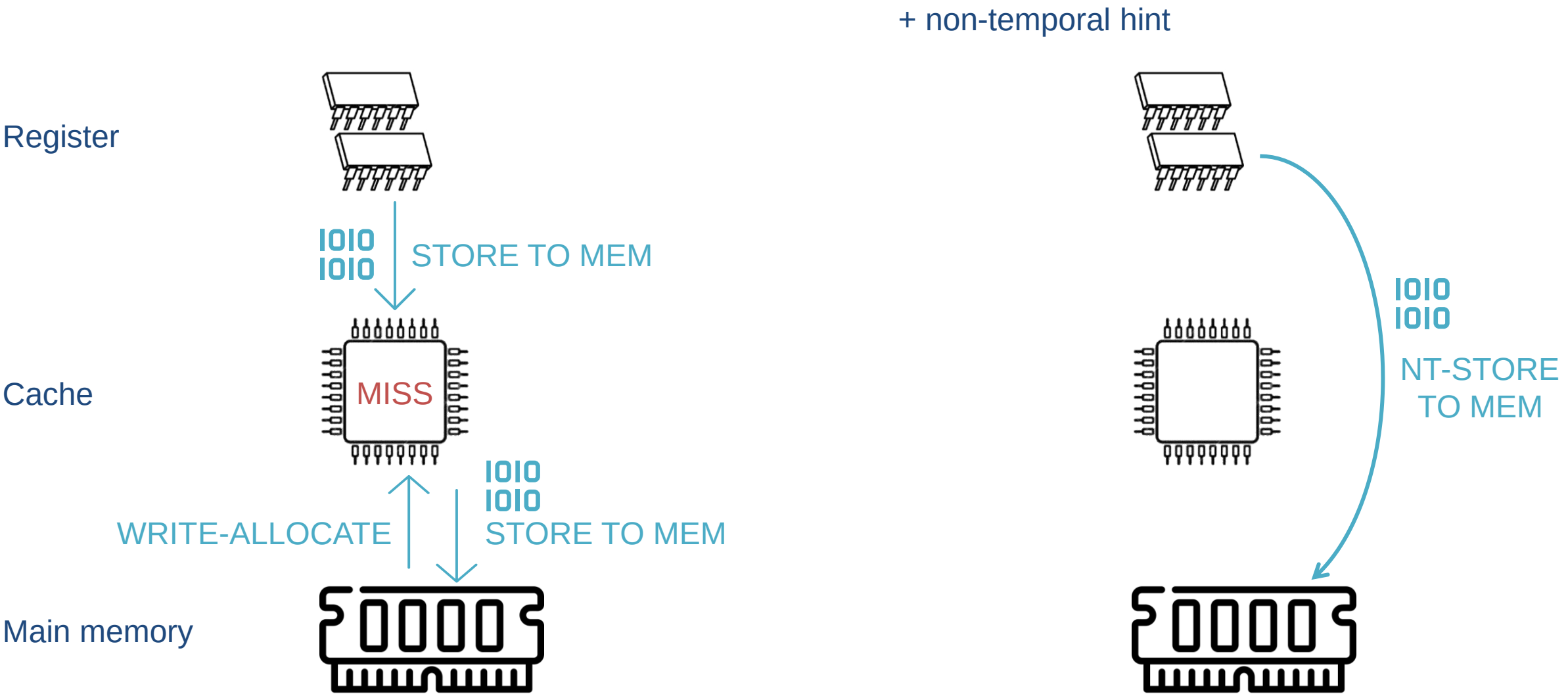
CPU clock frequencies

- SIMD-heavy code is power-intensive and hot

  → downclocking when using multiple cores

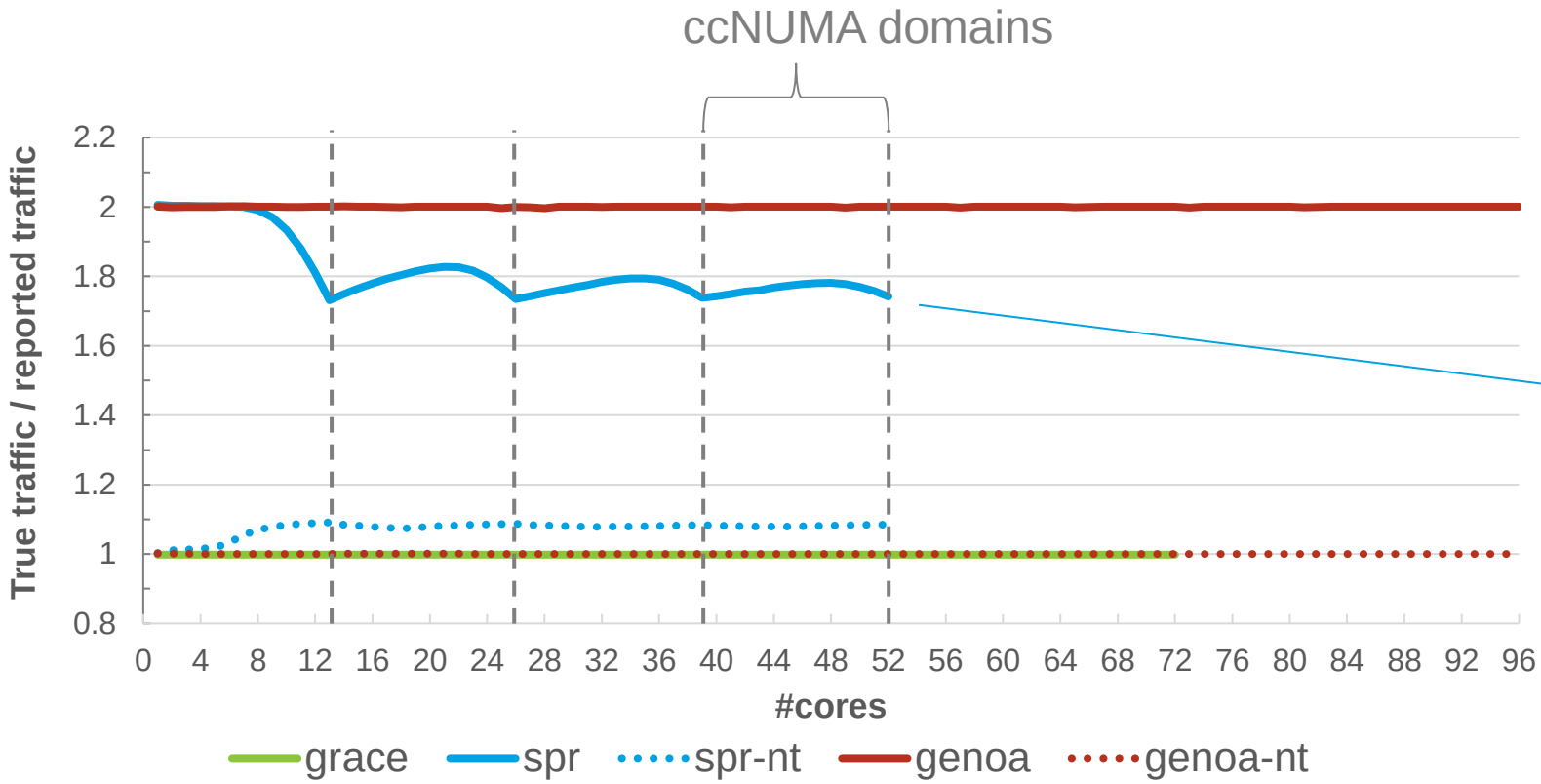


Write-allocate evasion



Write-allocate evasion

- Store-only benchmark : **`a[] = s`**
  - if **every** STORE triggers WA → 2
  - if **no** STORE triggers WA → 1



SpecI2M

For more see 10.1109/IPDPS57955.2024.00038

Summary & Outlook

- In-depth comparison between NVIDIA Grace CPU Superchip, Intel Sapphire Rapids, and AMD Genoa
  - Accurate in-core performance models
    - superior over other static code analyzer performance models, e.g., LLVM-MCA
  - Analysis of the sustained clock frequency for SIMD-heavy codes
  - Analysis of write-allocate evasion
- **Future work**
  - Extend work to a node-level performance model (Execution-Cache-Memory model)
  - Investigate server capabilities/peculiarities in real-life applications

References

[1] NVIDIA Grace Performance Tuning Guide

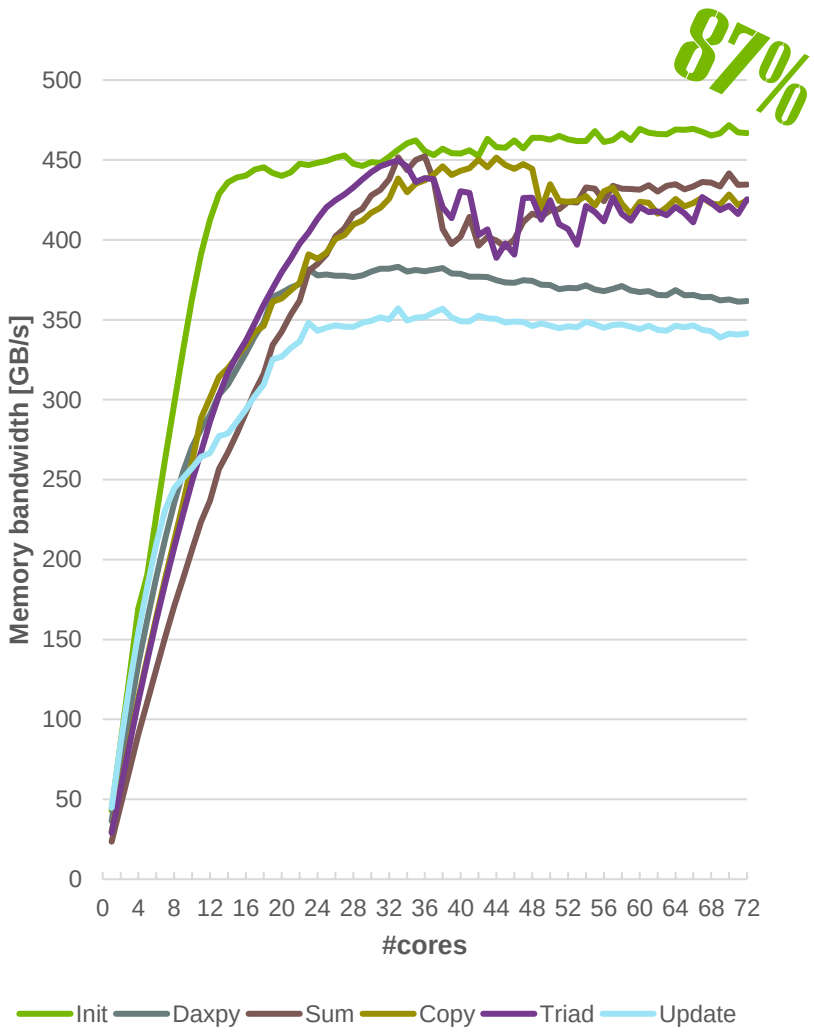
[2] https://wccftech.com/intel-sapphire-rapids-xeon-cpu-production-woes-moves-launch-to-early-2023/

[3] https://www.phoronix.com/review/amd-epyc-9684x-benchmarks

Backup – Memory Bandwidth

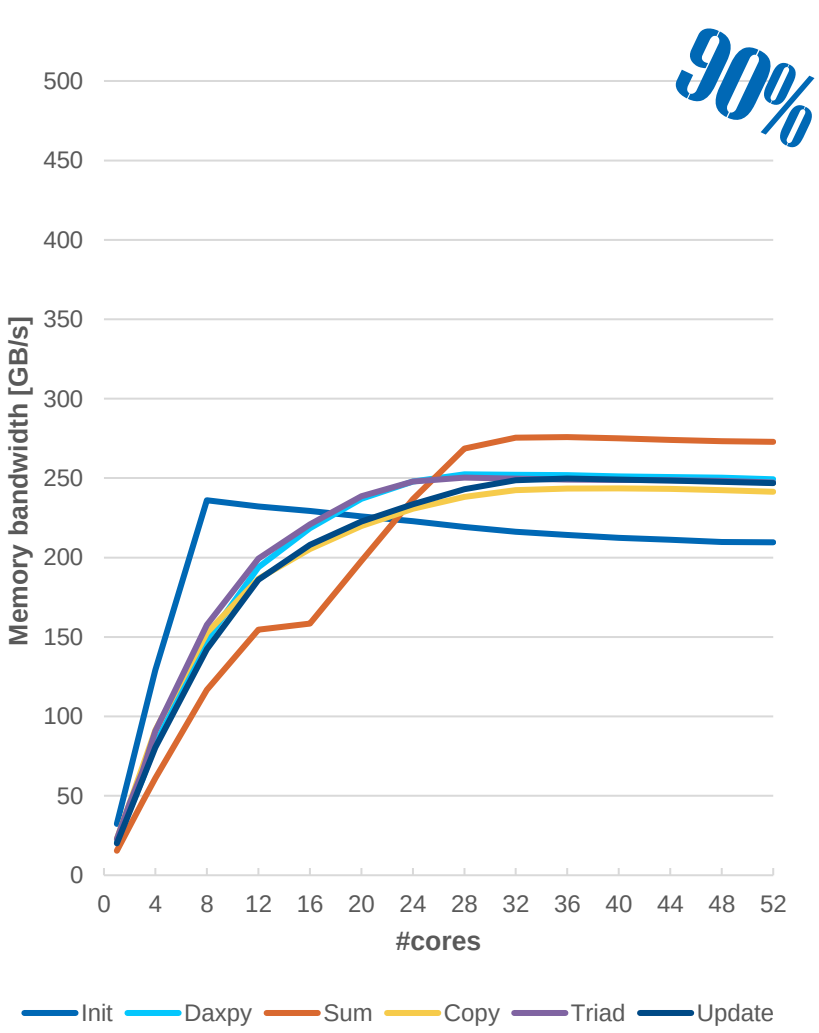
NVIDIA

Theor. max: 546 GB/s



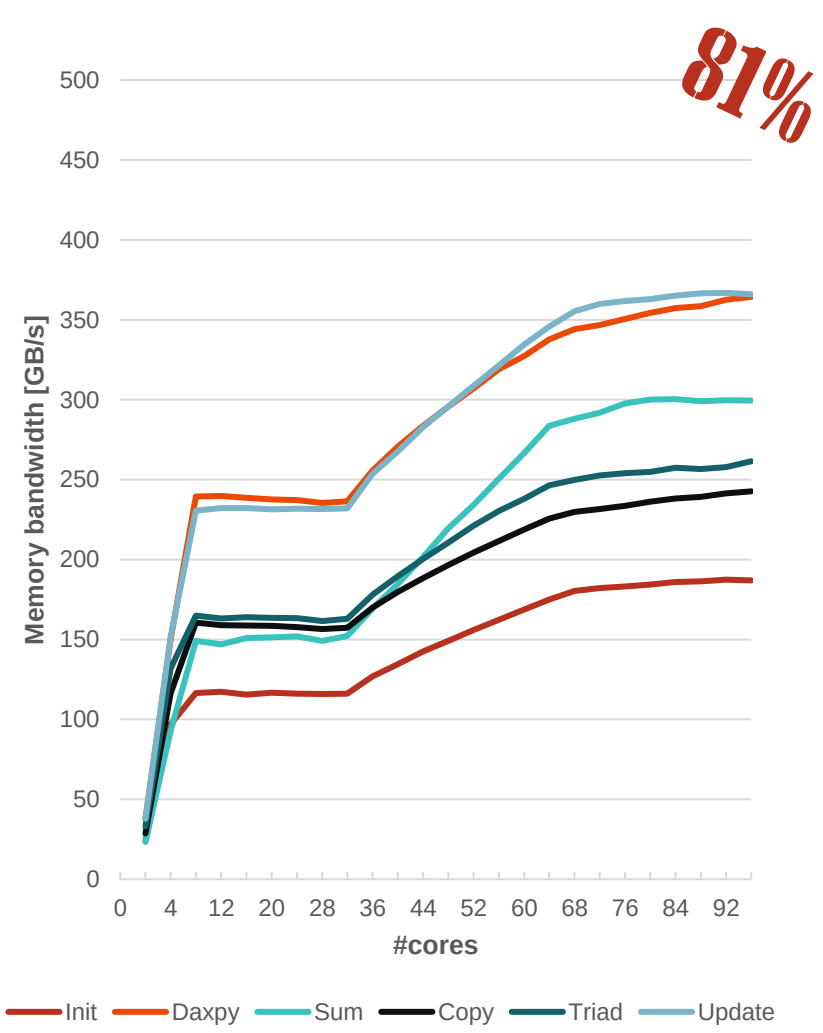
intel

Theor. max: 307 GB/s

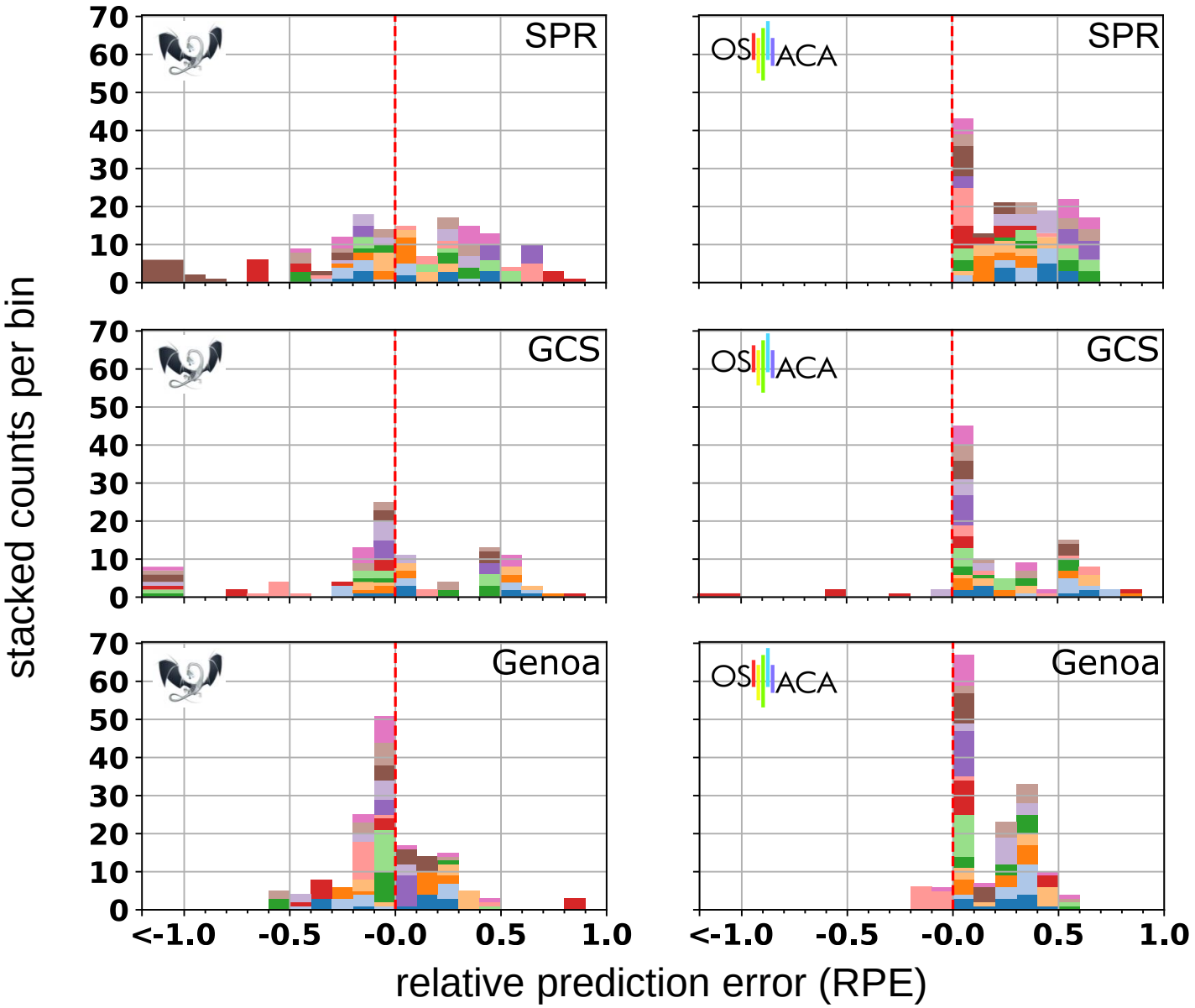


AMD

Theor. max: 461 GB/s



Model validation



OSACA

- 4% of all kernels over-predicted
- 37% within RPE ≤ 10%
- 44% within RPE ≤ 20%

LLVM-MCA

- 25% of all kernels over-predicted
- 10/16% within RPE ≤ 10/20%