Erlangen Regional Computing Center

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

The National High-Performance Computing Alliance – advancing the German HPC ecosystem

International Workshop on the Integration of (S+D+L) : Toward Society 5.0

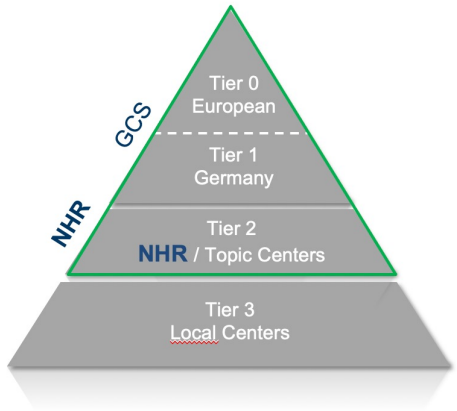
Prof. Dr. Gerhard Wellein
Erlangen Center for National HPC (NHR@FAU)
National HPC Alliance (NHR Verein)

NHR FAU

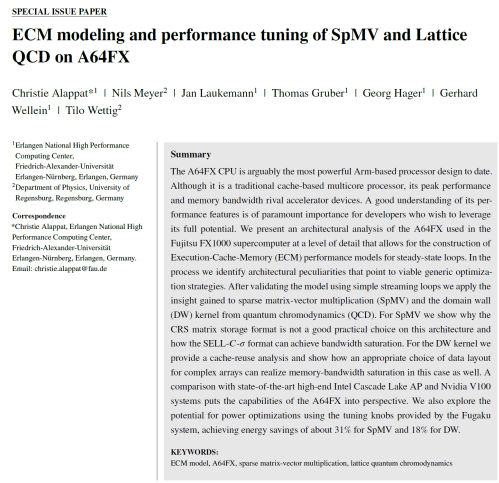
NHR

HPC High Performance Computing

- The NHR Alliance



- NHR Technology Evaluation: A64FX



SPECIAL ISSUE PAPER

ECM modeling and performance tuning of SpMV and Lattice QCD on A64FX

Christie Alappat*[1] | Nils Meyer[2] | Jan Laukemann[1] | Thomas Gruber[1] | Georg Hager[1] | Gerhard Wellein[1] | Tilo Wettig[2]

[1]Erlangen National High Performance Computing Center, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
[2]Department of Physics, University of Regensburg, Regensburg, Germany

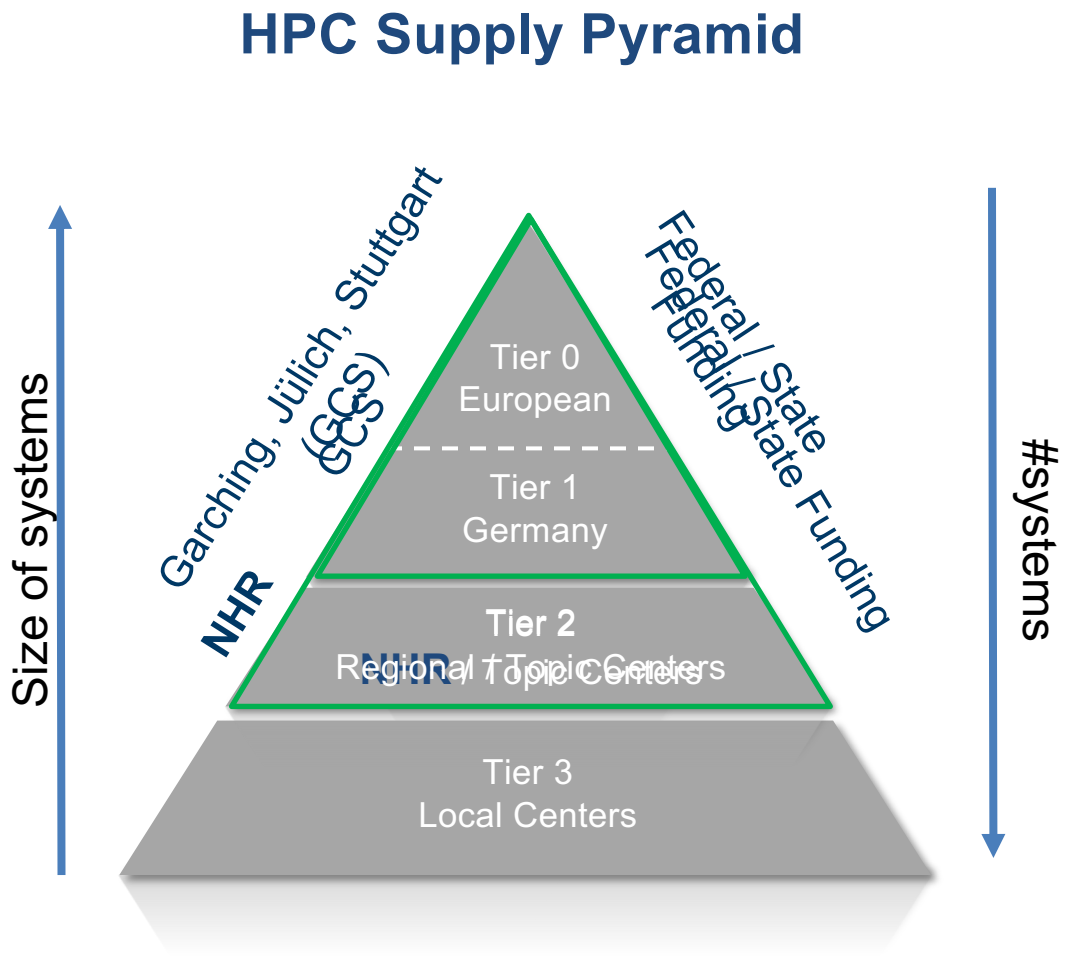
Correspondence
*Christie Alappat, Erlangen National High Performance Computing Center, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany.
Email: christie.alappat@fau.de

Summary

The A64FX CPU is arguably the most powerful Arm-based processor design to date. Although it is a traditional cache-based multicore processor, its peak performance and memory bandwidth rival accelerator devices. A good understanding of its performance features is of paramount importance for developers who wish to leverage its full potential. We present an architectural analysis of the A64FX used in the Fujitsu FX1000 supercomputer at a level of detail that allows for the construction of Execution-Cache-Memory (ECM) performance models for steady-state loops. In the process we identify architectural peculiarities that point to viable generic optimization strategies. After validating the model using simple streaming loops we apply the insight gained to sparse matrix-vector multiplication (SpMV) and the domain wall (DW) kernel from quantum chromodynamics (QCD). For SpMV we show why the CRS matrix storage format is not a good practical choice on this architecture and how the SELL-C-σ format can achieve bandwidth saturation. For the DW kernel we provide a cache-reuse analysis and show how an appropriate choice of data layout for complex arrays can realize memory-bandwidth saturation in this case as well. A comparison with state-of-the-art high-end Intel Cascade Lake AP and Nvidia V100 systems puts the capabilities of the A64FX into perspective. We also explore the potential for power optimizations using the tuning knobs provided by the Fugaku system, achieving energy savings of about 31% for SpMV and 18% for DW.

KEYWORDS:
ECM model, A64FX, sparse matrix-vector multiplication, lattice quantum chromodynamics

NHR as part of HPC in Germany



- A coordinated network of **National High Performance Computing (NHR)** centers at German universities funded by federal and state governments
- The NHR network shall also strengthen **methodological competence** through coordinated **training** and continuing **education** of users and, in particular, of young scientists
- Total funding **625M Euro** (2021-2030)
- Currently **8 NHR centers**
- **Official start: Jan 1st, 2021**

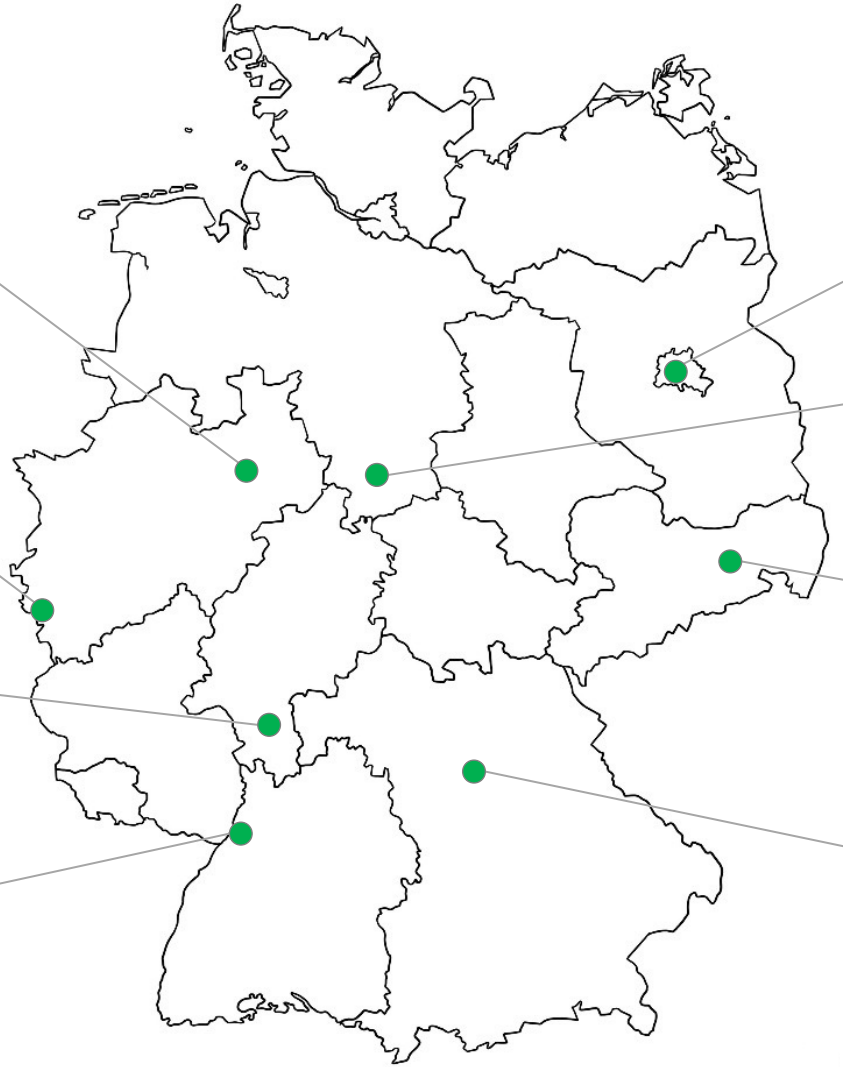
NHR Centers – open for Science & Research in Germany

Paderborn Center for Parallel Computing

NHR4CES RWTH AACHEN UNIVERSITY

NHR4CES TECHNISCHE UNIVERSITÄT DARMSTADT

KIT | NHR



NHR @ ZIB ZUSE INSTITUTE BERLIN

GWDG Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

ZIH Zentrum für Informationsdienste und Hochleistungsrechnen

NHR FAU

The whole is greater than the sum of the parts!

NHR Alliance – Mission & Goals

- Establish alliance of HPC-centers at **German universities**
- Focus on **demand of science** and **research**
- Support and drive **scientific developments**

Tasks

- Coordinated compute capabilities
- High quality user support
- Broad training program
- Cover all relevant application-/ HPC-fields

Goals

- Advance & broaden HPC-expertise
- Foster scientific computing
- Promote young scientists

NHR Alliance – Services & Activities: Compute time

- NHR resources are **open to researchers at German universities**
- Researchers may chose center which meets their requirements best
- **Application** for resources through **central portal (**available in Q2/2022)
- **Peer review process** – scientific quality / need for Tier-2 ressources
- Simplified review process for projects reviewed by DFG (and others)
- User/project support available through NHR-centers

NHR Alliance – Central legal entity

Legal entity has been established (NHR-Verein)

- NHR-Verein coordinates
  - financial planning / investments of NHR centers
  - training & teaching activities
  - central application portal for compute time and services

https://www.nhr-gs.de

- NHR-Verein supports
  - joint activties of the centers reaching beyond NHR
  - activities to foster scientific computing and young researchers

- Sign up for our mailing list:
  https://www.nhr-gs.de/aktuelles/veranstaltungen

NHR Alliance – Graduate School

Who? Students with a **masters degree** in Computer Science, Mathematics, the Natural or Engineering Sciences or equivalent degree

What? Education in (1) Operation & computer architecture, (2) Software & HPC methods, (3) Application of HPC methods, soft skill seminars, mentoring program

NHR scholarships: Up to nine PhD scholars/year, 2200 euros/month (tax-free), granted for 36 months, Start: April 1st 2022.

First application deadline: December 15th 2021

For more details see:
https://www.nhr-gs.de/ueber-uns/nhr-graduiertenschule

NHR Alliance – Selected central / joint activities

- Regular NHR PerfLab Seminar (online) – selected talks
  - Feb. 23.: G. Hager, NHR@FAU:
    A closer look at the Fujitsu A64FX processor
  - Dec.15.: J. McCalpin, TACC:
    Memory Bandwidth and System Balance in HPC Systems – 2021 Update
  - See: https://hpc.fau.de/category/all/hpc/nhr-perflab-seminar/

- (Future) Technology Evaluation – e.g. A64FX
  - NHR@FAU: Performance Modeling / Engineering
  - NHR@KIT: HPE Apollo 80 System – 8 A64FX
  - Univ. of Regensburg (Physics): HPE / CRAY - 64 A64FX + 100 Gbit EDR

  + Fugaku + Ookami (BNL / Stony Brook)

NHR FAU

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

NHR Technology Evaluation

Modeling and tuning of SpMV and a lattice QCD kernel on the A64FX

Joint work wit N. Meyer &T. Wettig
(QCD, Department of Physic, U. Regensburg, Germany)

Paper

C. Alappat, N. Meyer, J. Laukemann, T. Gruber, G. Hager, G. Wellein, and T. Wettig:

ECM modeling and performance tuning of SpMV and Lattice QCD on A64FX.

Concurrency and Computation: Practice and Experience, e6512 (2021).

Available with Open Access.

DOI: 10.1002/cpe.6512

SPECIAL ISSUE PAPER

ECM modeling and performance tuning of SpMV and Lattice QCD on A64FX

Christie Alappat*[1] | Nils Meyer[2] | Jan Laukemann[1] | Thomas Gruber[1] | Georg Hager[1] | Gerhard Wellein[1] | Tilo Wettig[2]

[1]Erlangen National High Performance Computing Center, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
[2]Department of Physics, University of Regensburg, Regensburg, Germany

Correspondence
*Christie Alappat, Erlangen National High Performance Computing Center, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany.
Email: christie.alappat@fau.de

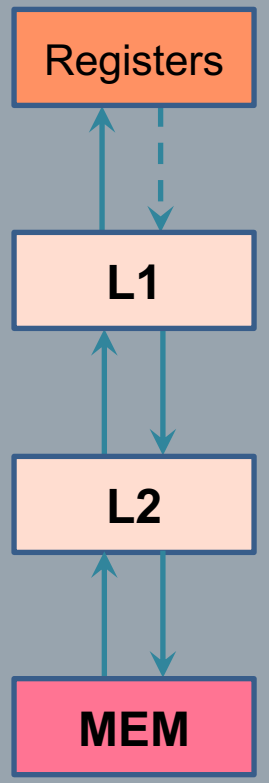
Summary

The A64FX CPU is arguably the most powerful Arm-based processor design to date. Although it is a traditional cache-based multicore processor, its peak performance and memory bandwidth rival accelerator devices. A good understanding of its performance features is of paramount importance for developers who wish to leverage its full potential. We present an architectural analysis of the A64FX used in the Fujitsu FX1000 supercomputer at a level of detail that allows for the construction of Execution-Cache-Memory (ECM) performance models for steady-state loops. In the process we identify architectural peculiarities that point to viable generic optimization strategies. After validating the model using simple streaming loops we apply the insight gained to sparse matrix-vector multiplication (SpMV) and the domain wall (DW) kernel from quantum chromodynamics (QCD). For SpMV we show why the CRS matrix storage format is not a good practical choice on this architecture and how the SELL-C-σ format can achieve bandwidth saturation. For the DW kernel we provide a cache-reuse analysis and show how an appropriate choice of data layout for complex arrays can realize memory-bandwidth saturation in this case as well. A comparison with state-of-the-art high-end Intel Cascade Lake AP and Nvidia V100 systems puts the capabilities of the A64FX into perspective. We also explore the potential for power optimizations using the tuning knobs provided by the Fugaku system, achieving energy savings of about 31% for SpMV and 18% for DW.

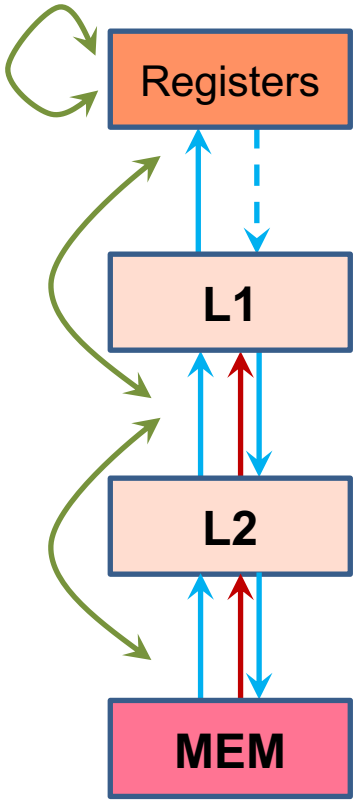
KEYWORDS:
ECM model, A64FX, sparse matrix-vector multiplication, lattice quantum chromodynamics

Single-core analysis

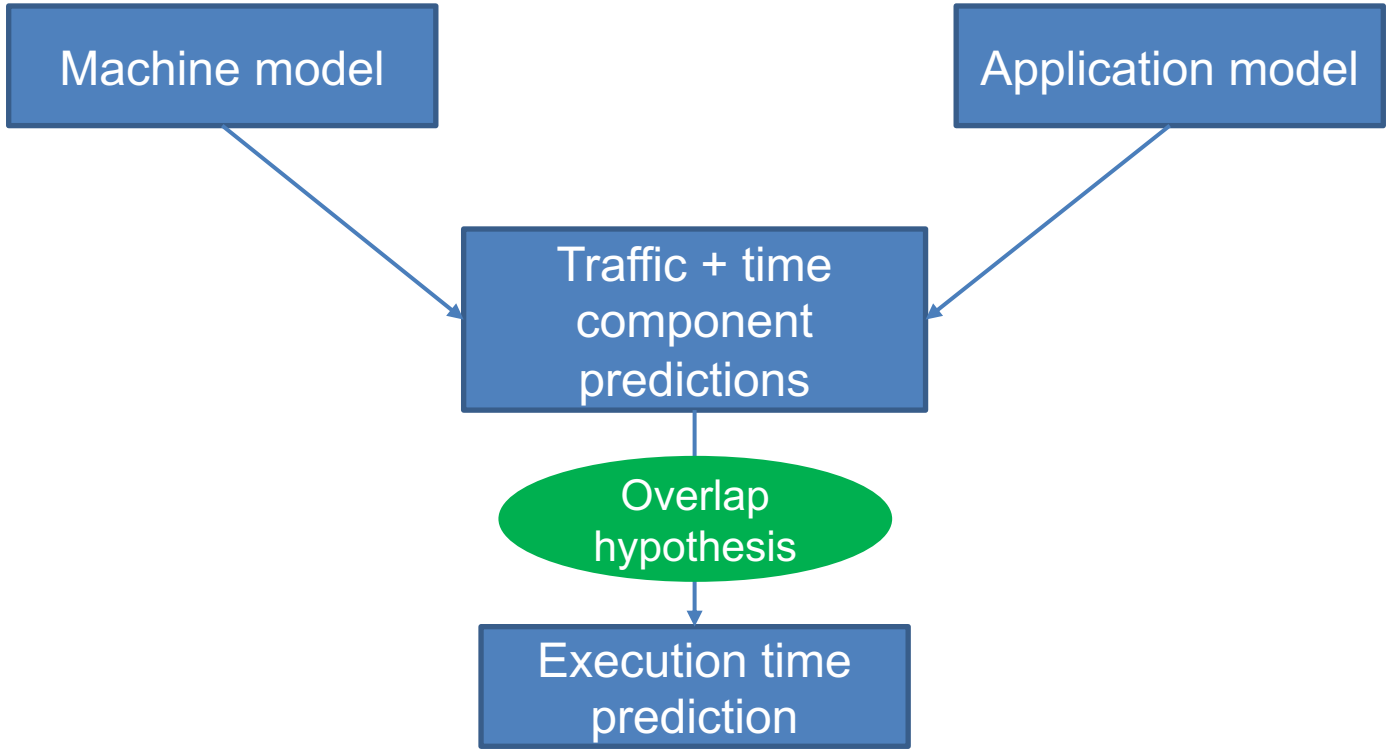
ECM model



Single-core ECM model



Execution-Cache-Memory (ECM) model helps us to understand and analyze the single-core performance.



Hofmann et.al.: *Bridging The Architecture Gap: Abstracting Performance-Relevant Properties Of Modern Server Processors*, https://doi.org/10.14529/jsfi200204

In-core prediction

Application knowledge

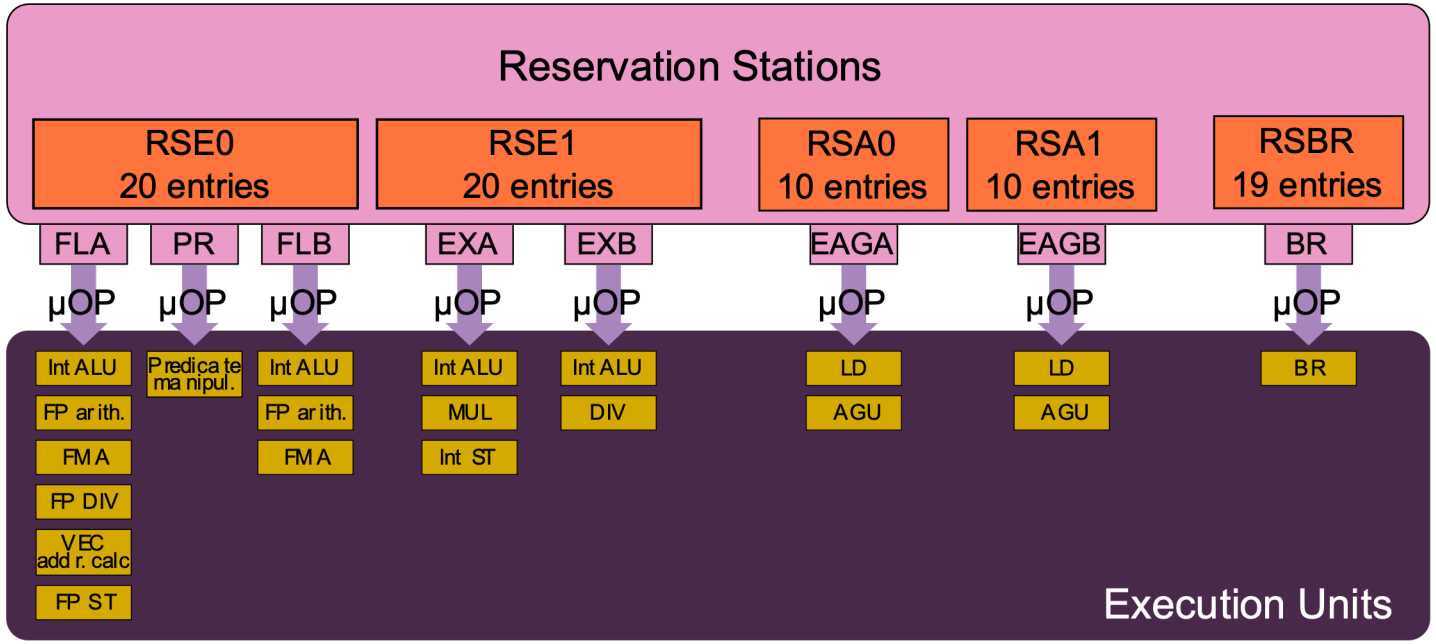
STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
.L18:
  ld1d z4.d, p5/z, [x21, x9, lsl 3]
  ld1d z5.d, p5/z, [x20, x9, lsl 3]
  fmad z5.d, p5/m, z2.d, z4.d
  st1d z5.d, p5, [x19, x9, lsl 3]
  add x8, x9, 8
  whilelo p5.d, w8, w7
  b.any .L18
```

2cy / VL

Machine knowledge



In-core prediction

Application knowledge

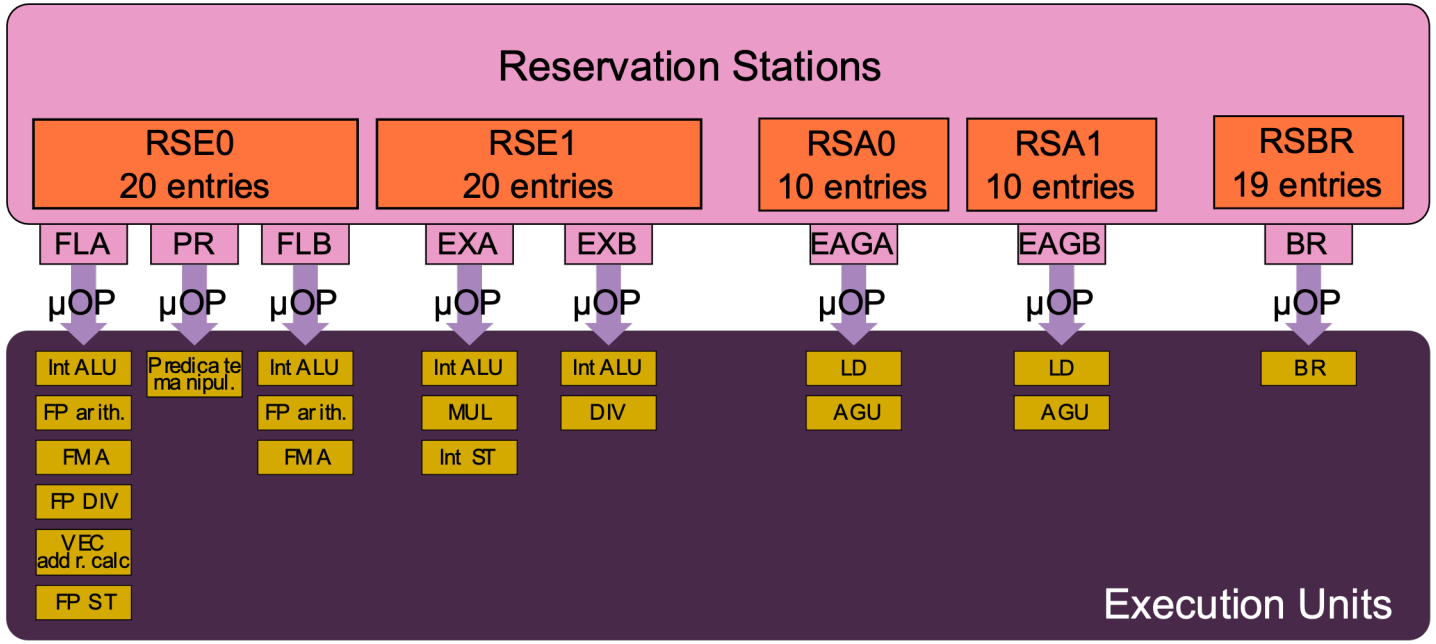
STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
.L18:
 ld1d z4.d, p5/z, [x21, x9, lsl 3]
 ld1d z5.d, p5/z, [x20, x9, lsl 3]
 fmad z5.d, p5/m, z2.d, z4.d
 st1d z5.d, p5, [x19, x9, lsl 3]
 add x8, x9, 8
 whilelo p5.d, w8, w7
 b.any .L18
```

2cy / VL

Machine knowledge



Static analysis and prediction of in-core contribution

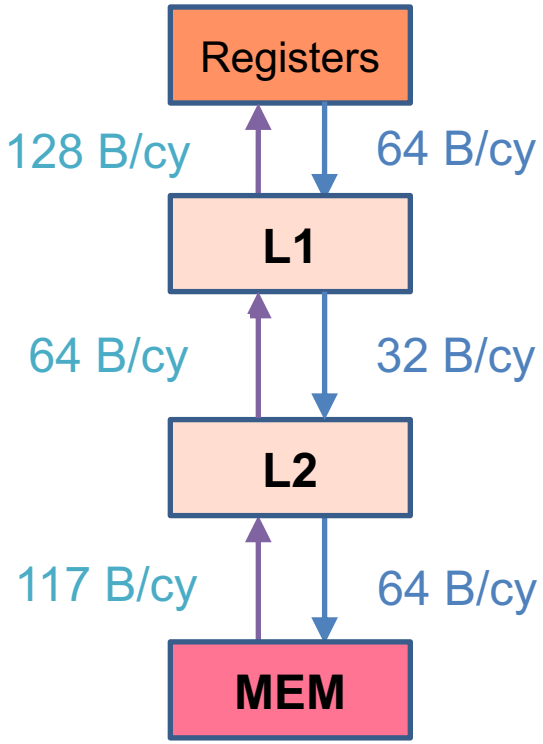
OSACA

https://github.com/RRZE-HPC/OSACA

available for A64FX

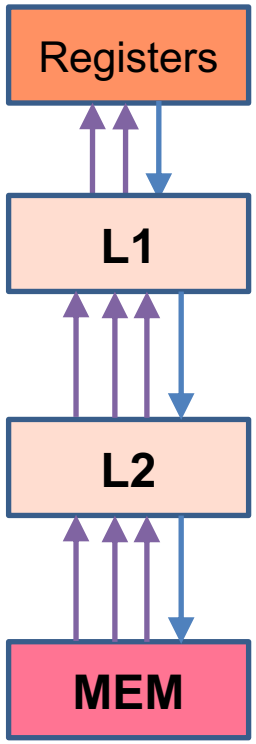
Data transfer for STREAM triad

Machine knowledge
(FX700)

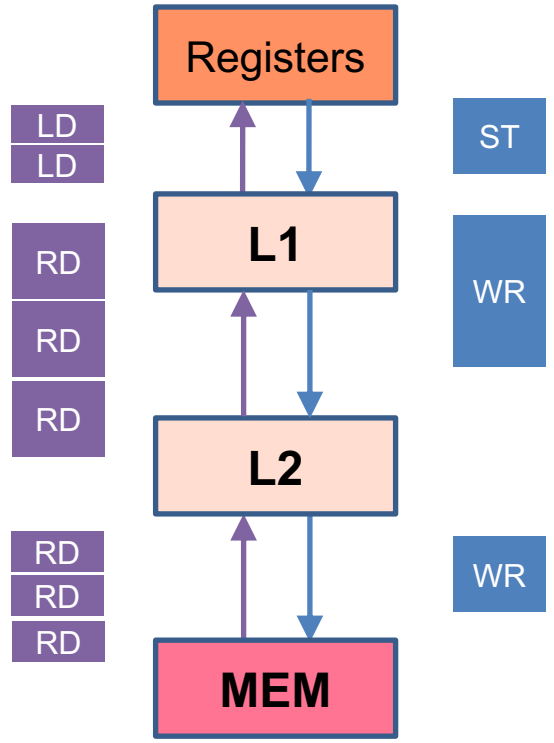


Application knowledge
STREAM triad

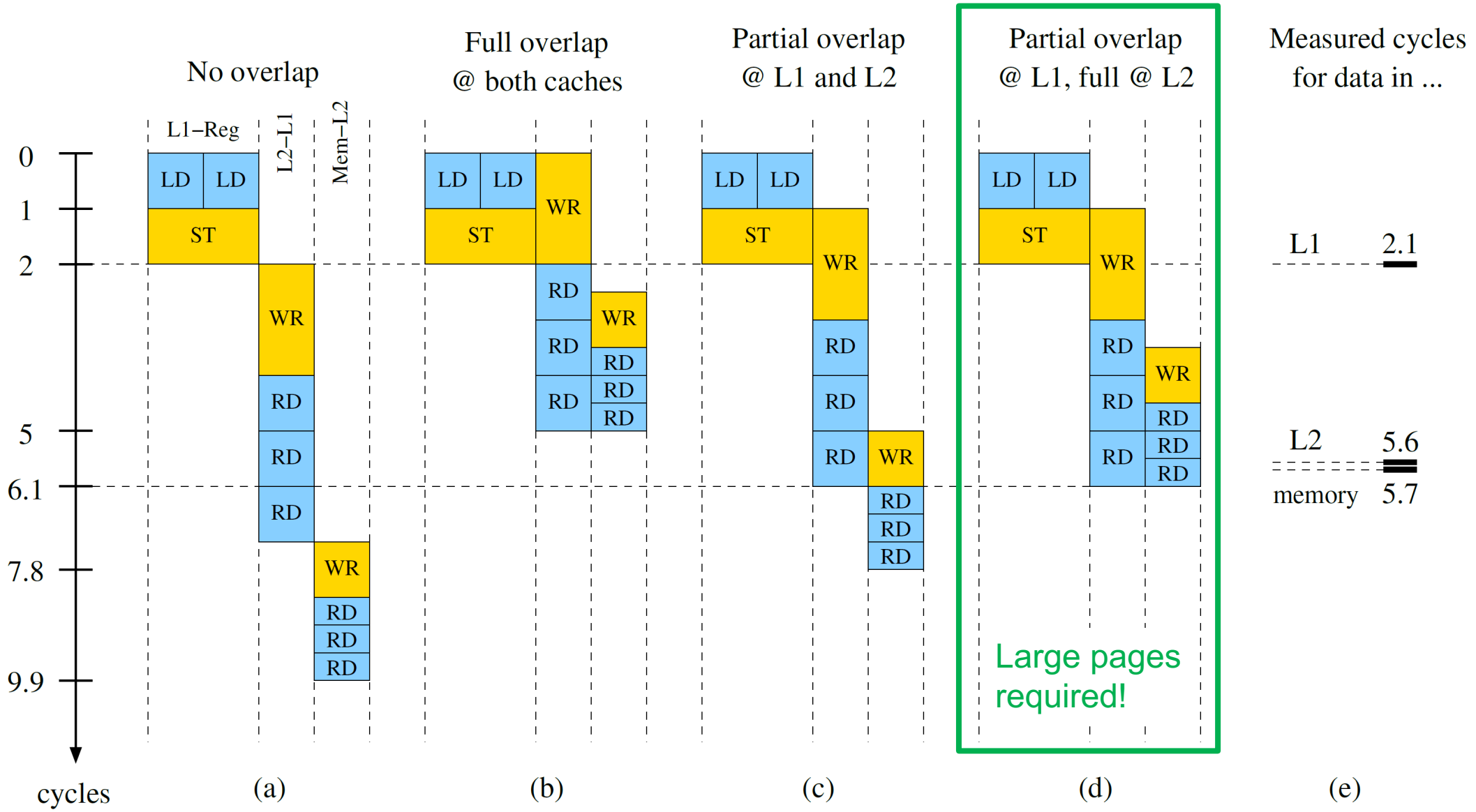
```
a[i] = b[i] + s*c[i]
```



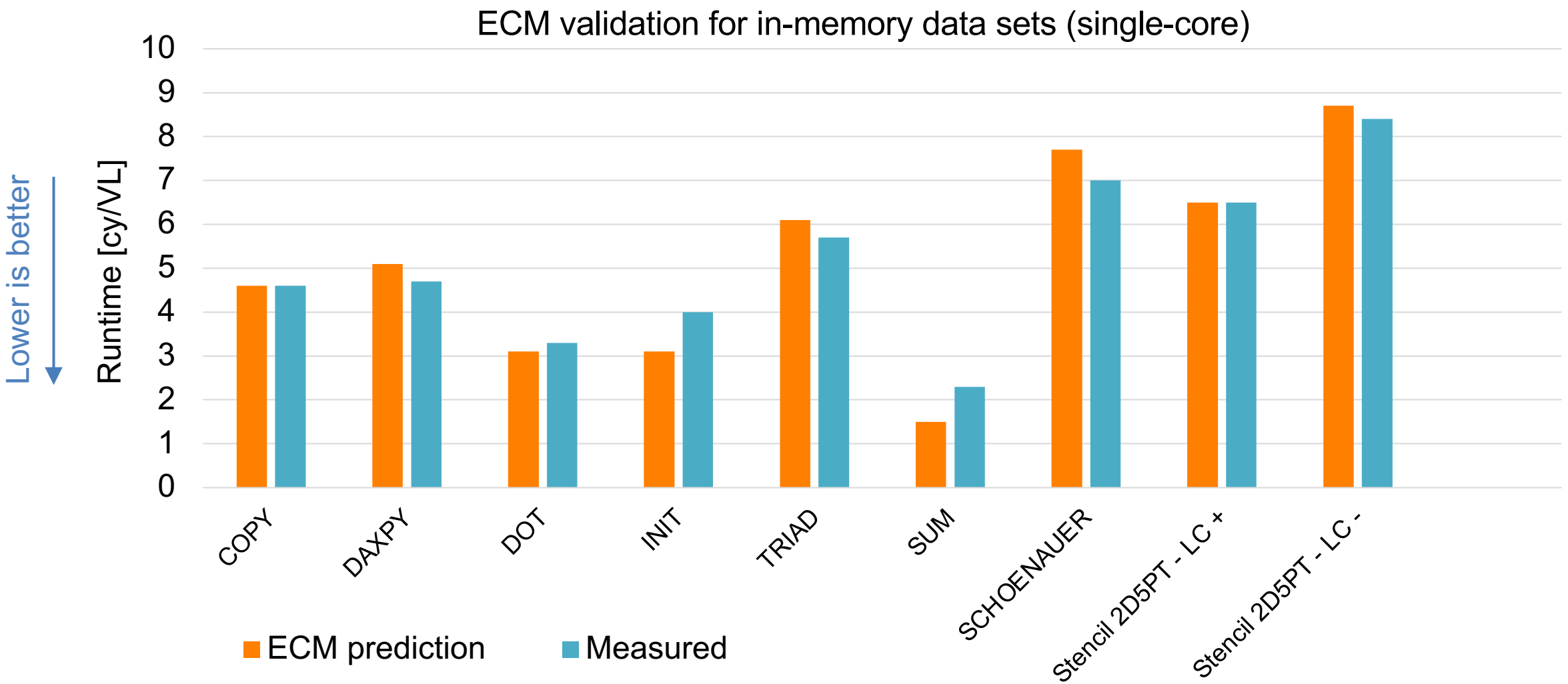
ECM prediction?
STREAM triad on A64FX



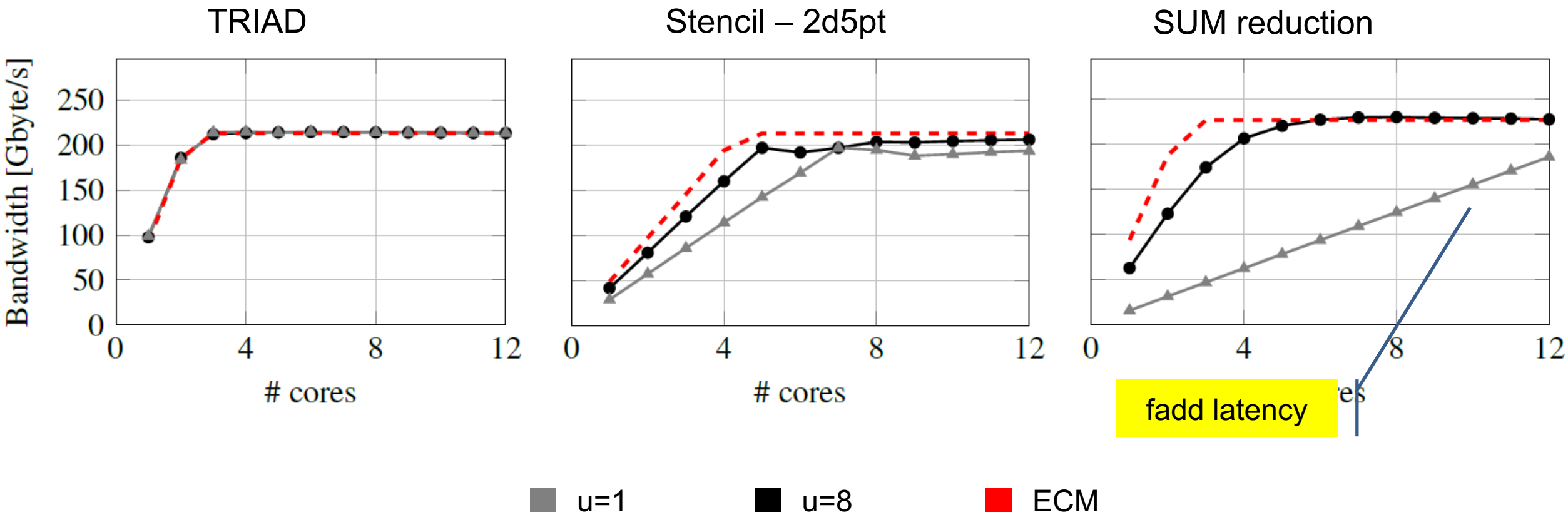
Overlap hypotheses for A64FX



Model validation (FX1000, large pages)



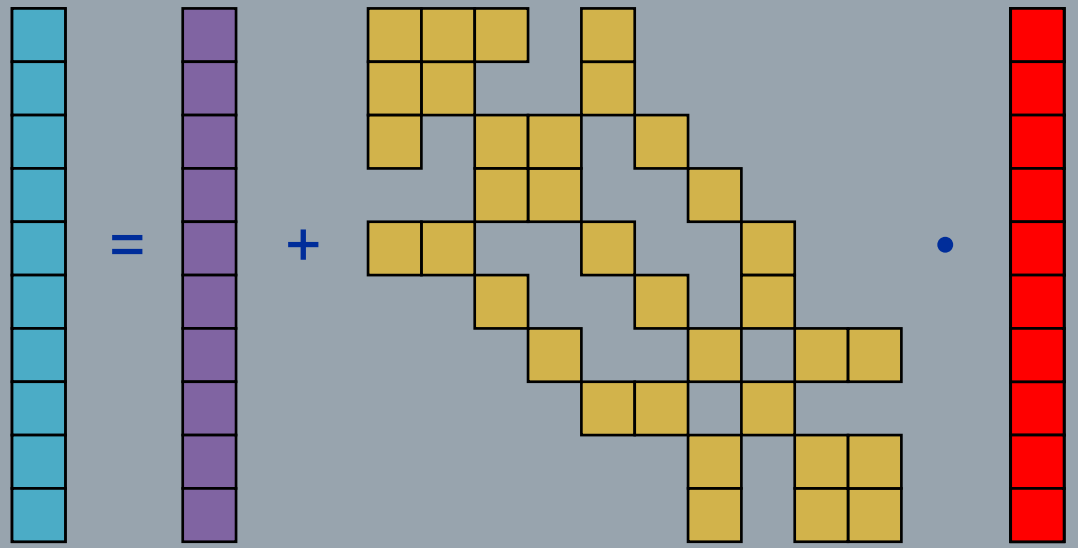
Multicore (in-memory data set)



Sufficient unrolling is crucial (but sometimes it's not enough)

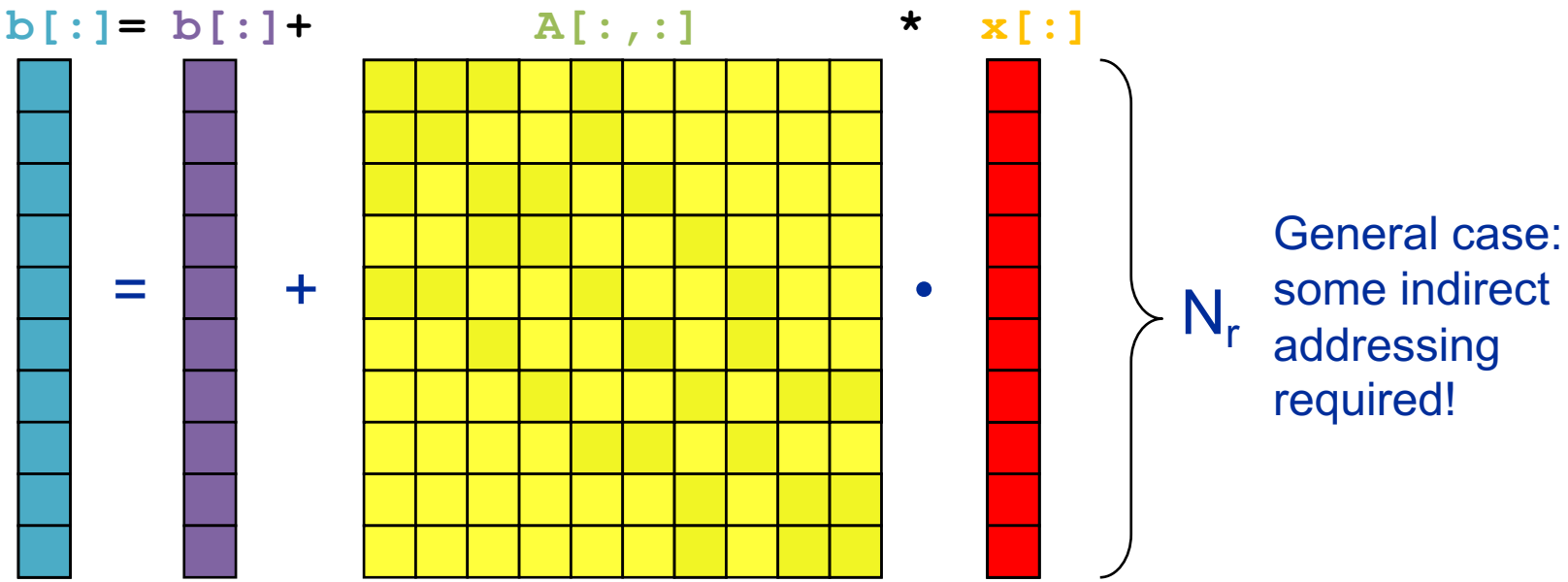
SpMV

Sparse Matrix-Vector Multiplication



SpMV

Sparse Matrix-Vector Multiplication (SpMV) : `b=Ax`



General case: some indirect addressing required!

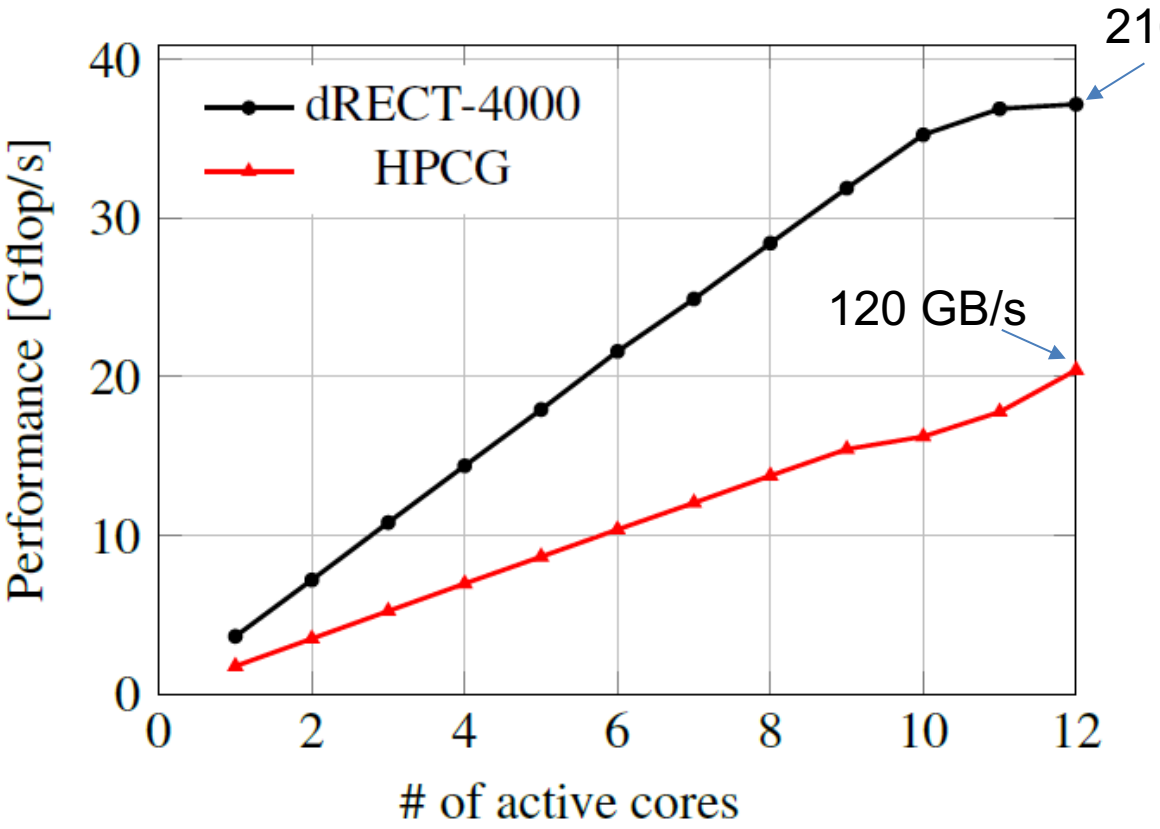
In Compressed Row Storage (CRS) format

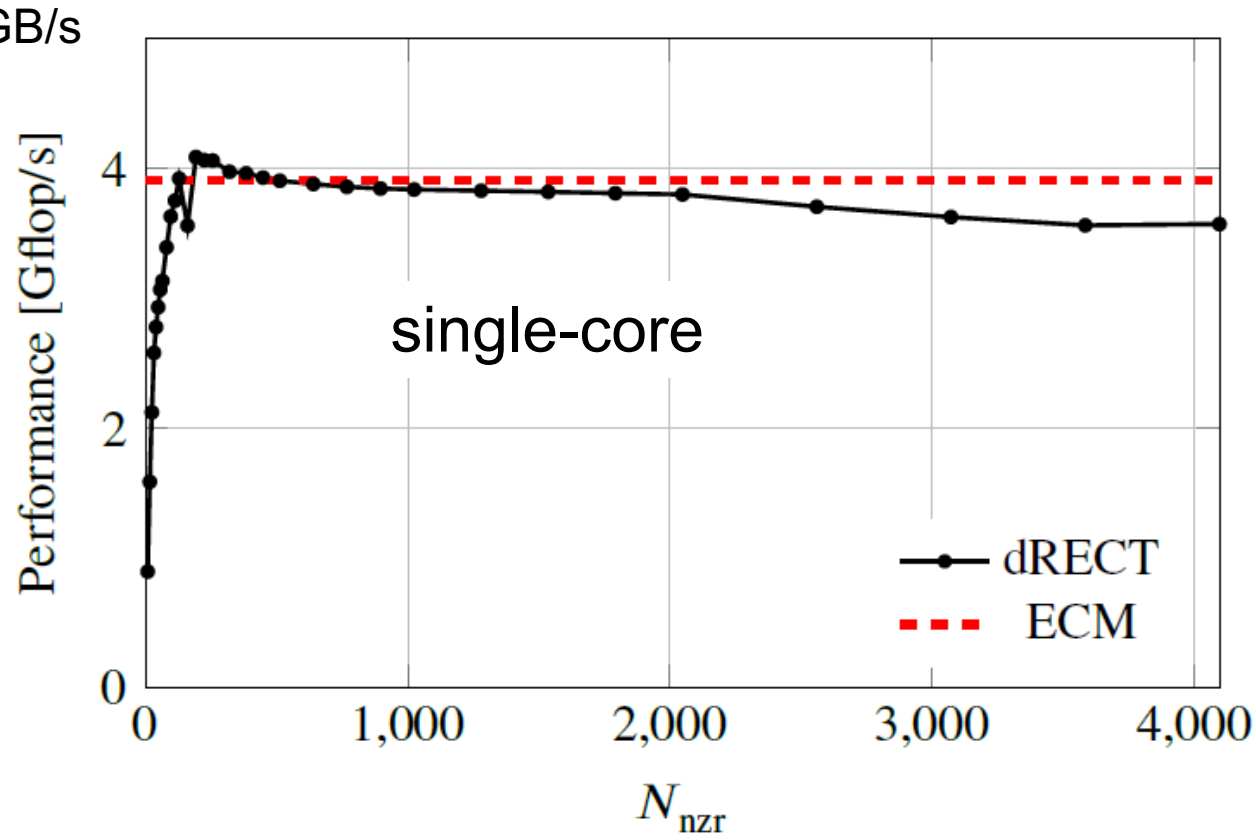
```
for i = 0:nrows-1 //Long outer loop
    for j = row_ptr[i]:row_ptr[i+1]-1 // Short inner loop
        b[i] = b[i] + A[j] * x[col_idx[j]]
```

Minimum code balance:

$$B_c^{min} = 6 \frac{\text{byte}}{\text{flop}}$$

SpMV – dRECT vs. HPCG-128^3





- dRECT: 4000-column tall & skinny dense matrix ($N_{nzr} = 4000$)
- HPCG: matrix from HPCG benchmark ($N_{nzr} = 27$), 128^3 rows

SpMV

Assembly of the short inner-loop

```
.L6:
        ld1sw     z0.d, p0/z, [x17, x20, lsl 2]
        ld1d      z2.d, p0/z, [x18, x20, lsl 3]
        ld1d      z3.d, p0/z, [x30, z0.d, lsl 3]
        add       x20, x20, 8
        fmla      z1.d, p0/m, z3.d, z2.d
        whilelo   p0.d, x20, x14
        b.any     .L6

        faddv     d4, p1, z1.d
```

FMA: Update **z1.d**

Latency: 9 cycles

Loop length : 27
HPCG matrix

Horizontal add of 512-bit register

latency = 49 cycles

ECM model predicts maximum bandwidth of 100 GB/s

→ No saturation

In Compressed Row Storage (CRS) format

```
for i = 0:nrows-1 //Long outer loop
    for j = row_ptr[i]:row_ptr[i+1]-1 // Short inner loop
        b[i] = b[i] + A[j] * x[col_idx[j]]
```

The problem with SpMV on A64FX

We need both:

- SIMD vectorization
- Modulo Variable Expansion (MVE)

With CRS, both must be implemented in the inner loop. The partial sums accumulation adds to the overhead.

Can we get rid of the partial sums accumulation *and* separate SIMD from MVE?

NHR FAU

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

CRS → SELL-C-σ

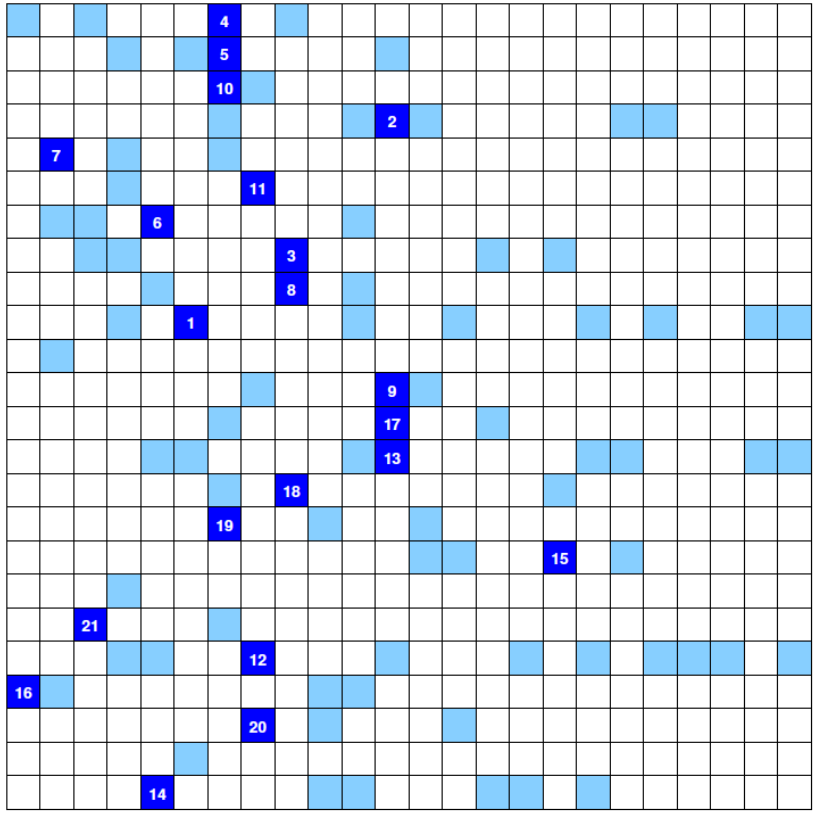
Change data storage format

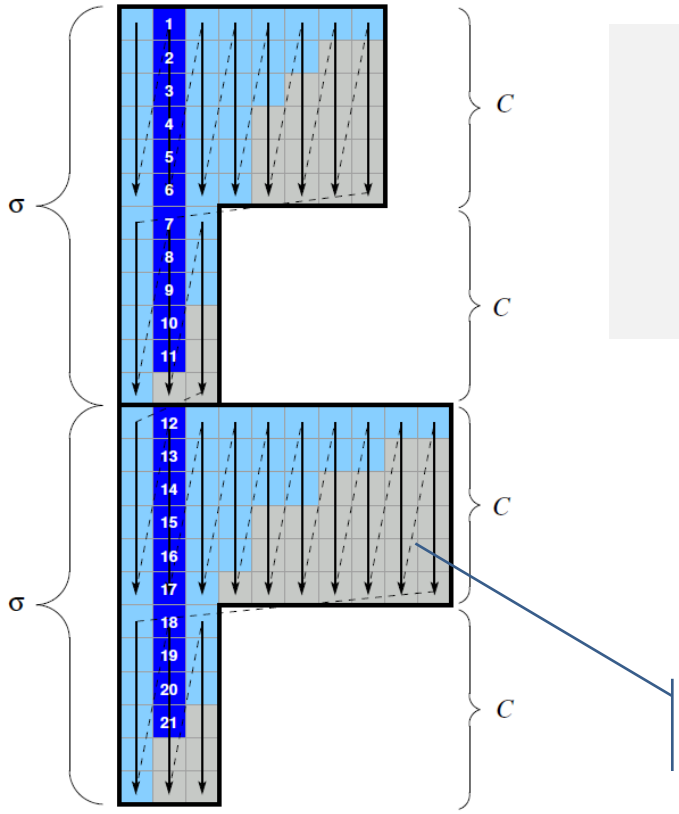
High Performance
Computing

SELL-C-σ

Idea

- Sort rows according to length within sorting scope σ
- Store nonzeros column-major in zero-padded blocks of height C





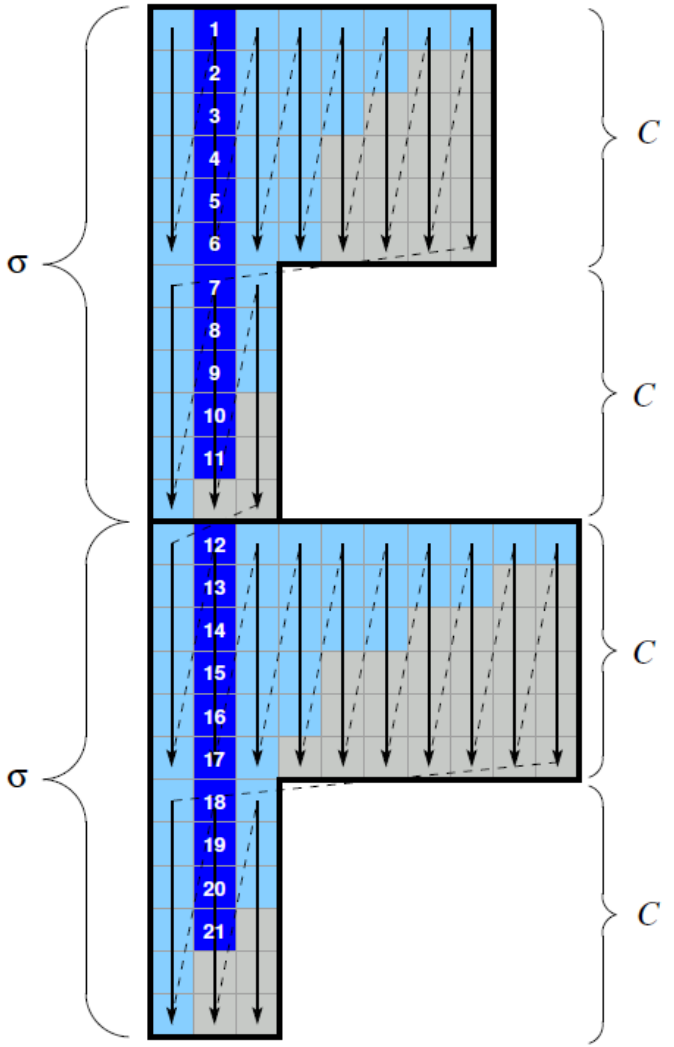
"Chunk occupancy":

$$\beta = \frac{N_{nz}}{\sum_{i=0}^{N_c} C \cdot l_i}$$

l_i: width of chunk i

How to choose the parameters?

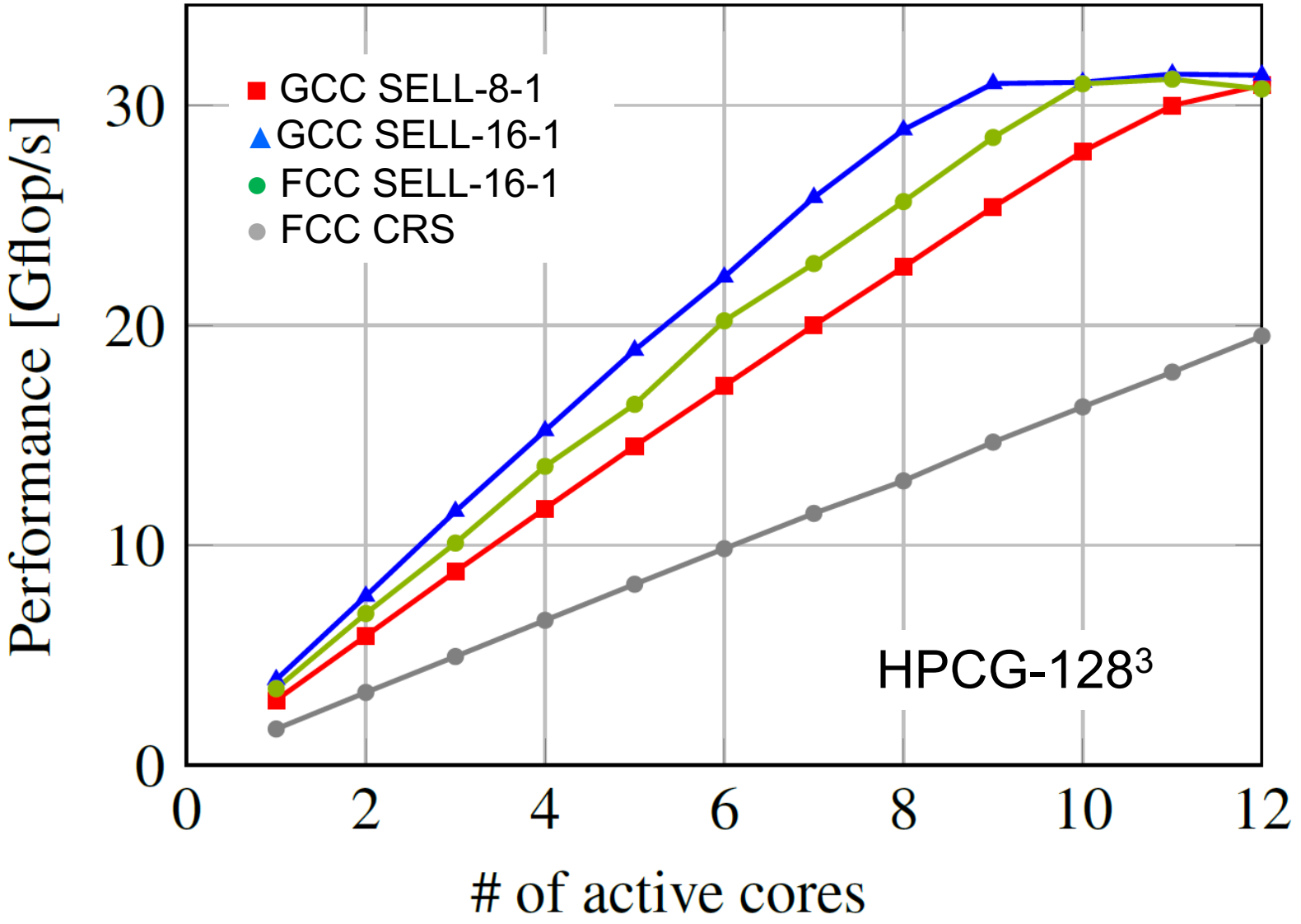
- C
  - $n\times$ SIMD width to allow good utilization of SIMD units
  - $n > 1$ useful for hiding ADD pipeline latency
- σ
  - As small as possible, as large as necessary
  - Large σ reduces zero padding (brings β closer to 1)
  - Sorting alters RHS access pattern → α depends on σ



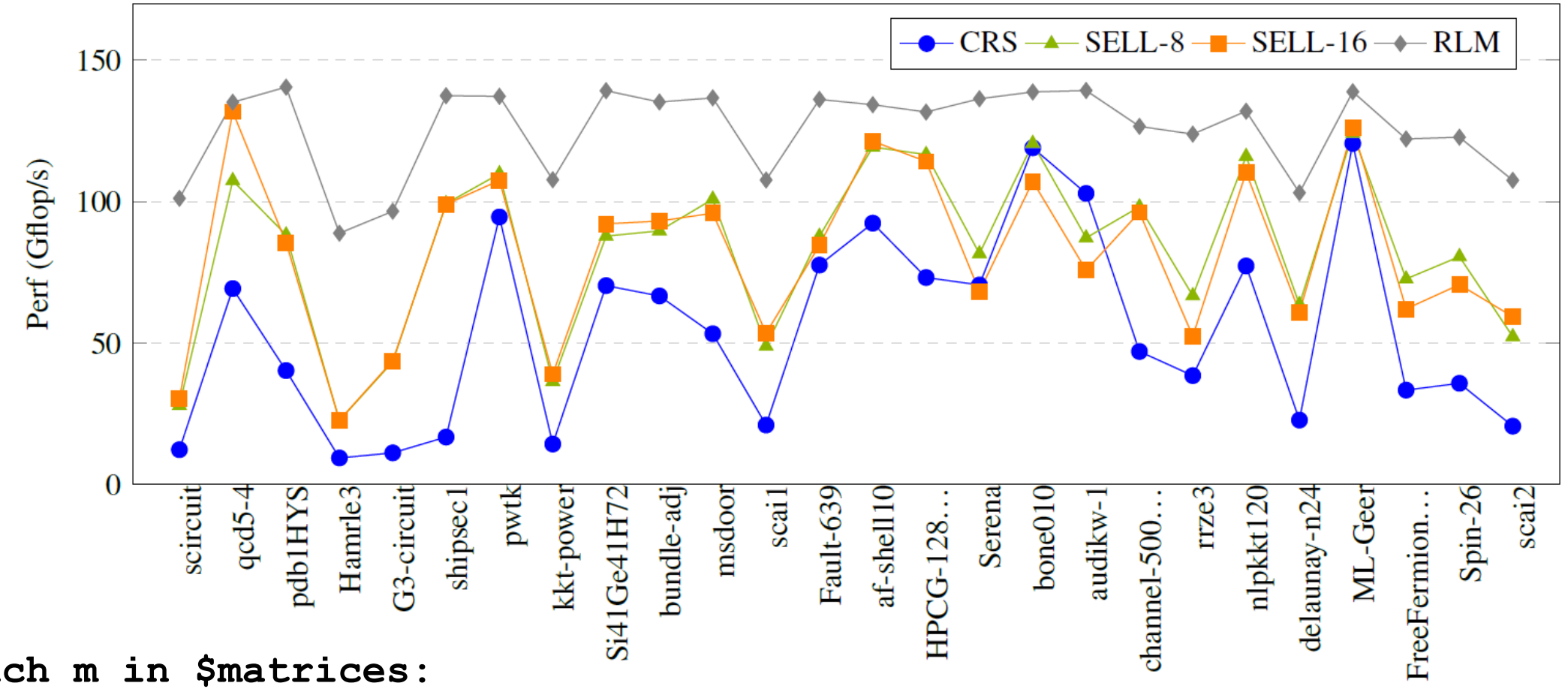
M. Kreutzer, G. Hager, G. Wellein, H. Fehske, and A. R. Bishop: *A unified sparse matrix data format for efficient general sparse matrix-vector multiplication on modern processors with wide SIMD units*. SIAM Journal on Scientific Computing **36**(5), C401–C423 (2014). DOI: 10.1137/130930352,

SpMV performance with SELL-C-σ (1 CMG)

- SELL-C-σ separates SIMD from sum reduction
- C>8 allows for reduction of fmla latency impact

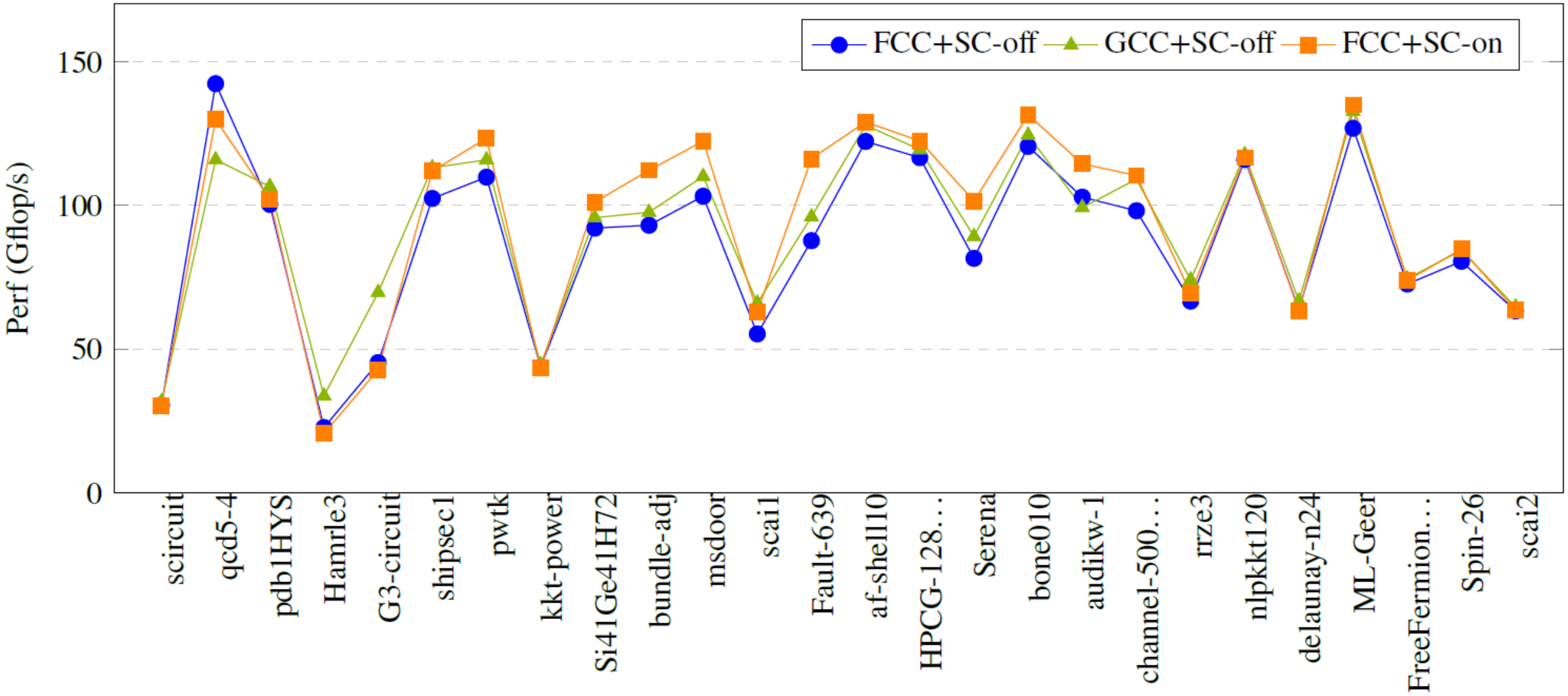


SpMV performance with SELL-C-σ (full chip)




```
foreach m in $matrices:
  apply RCM reordering if helpful
  try row-based vs. nonzero-based load balancing
  scan σ from 1 … 4096
```

Impact of Sector Cache



NHR FAU

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Domain Wall (DW) kernel

from Quantum Chromodynamics (QCD)

Context

- Lattice QCD simulates the strong interaction
- Iterative multigrid techniques on regular (4D or 5D) lattices
- Core component: Apply Dirac operator D to quark-field vector Ψ
- Domain Wall (DW) formulation: quark field lives on 4D boundary of a 5D space-time volume $V_4 \times L_s$

$$(D\psi)(n,s)_{\alpha a} =$$

$$\sum_{\mu=1}\sum_{\beta=1}\sum_{b=1}\left\{U_\mu(n)_{ab}(1+\gamma_\mu)_{\alpha\beta}\psi(n+\hat{\mu},s)_{\beta b} + U_\mu^\dagger(n-\hat{\mu})_{ab}(1-\gamma_\mu)_{\alpha\beta}\psi(n-\hat{\mu},s)_{\beta b}\right\}$$

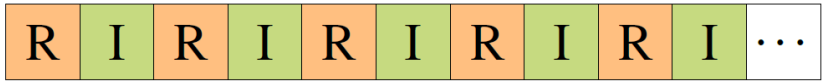
DW stencil kernel (simplified)

```
#define x_p 1 // x-plus  direction
#define x_m 2 // x-minus direction
#define y_p 3 // y-plus  direction
...
#pragma omp parallel for schedule(static)
for {t,z,y,x} = 1:{L_t-2,L_z-2,L_y-2,L_x-2}//collapsed loop over 4d space-time
{
  for(int s=0; s<L_s; ++s) //loop over 5th dimension
  {
    O[t][z][y][x][s] = R(x_p)·U[x_p][t][z][y][x]·P(x_p)·I[t][z][y][x+1][s] +
                       R(x_m)·U[x_m][t][z][y][x]·P(x_m)·I[t][z][y][x-1][s] +
                       R(y_p)·U[y_p][t][z][y][x]·P(y_p)·I[t][z][y+1][x][s] +
                       R(y_m)·U[y_m][t][z][y][x]·P(y_m)·I[t][z][y-1][x][s] +
                       R(z_p)·U[z_p][t][z][y][x]·P(z_p)·I[t][z+1][y][x][s] +
                       R(z_m)·U[z_m][t][z][y][x]·P(z_m)·I[t][z-1][y][x][s] +
                       R(t_p)·U[t_p][t][z][y][x]·P(t_p)·I[t+1][z][y][x][s] +
                       R(t_m)·U[t_m][t][z][y][x]·P(t_m)·I[t-1][z][y][x][s];
  }
}
```

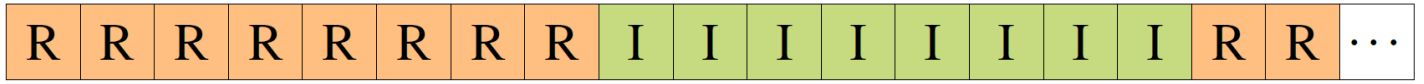
- "Grid" lattice QCD framework
- Uses SVE intrinsics
- Data type: double complex

Complex numbers data layout choice

RIRI (standard)



RRII

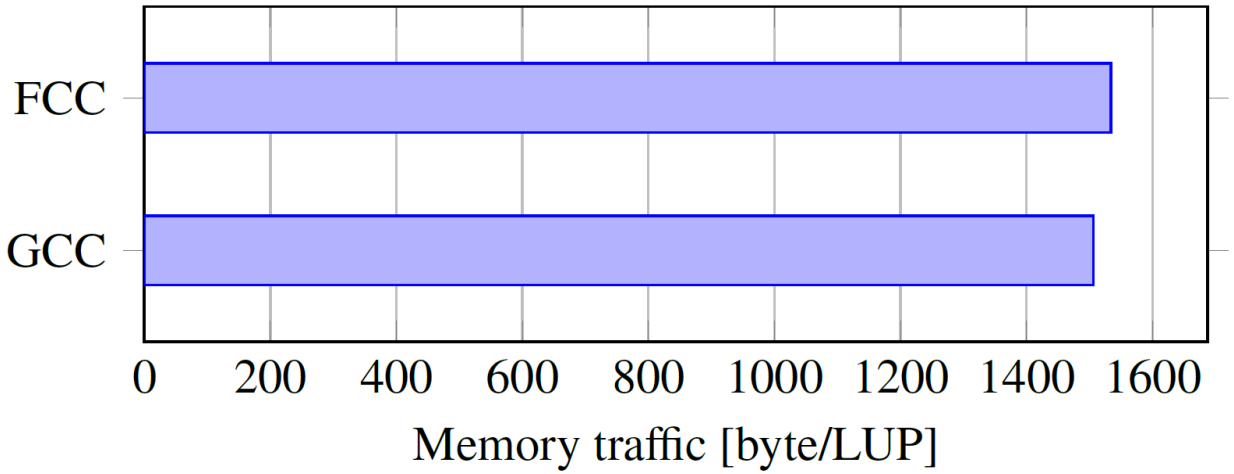


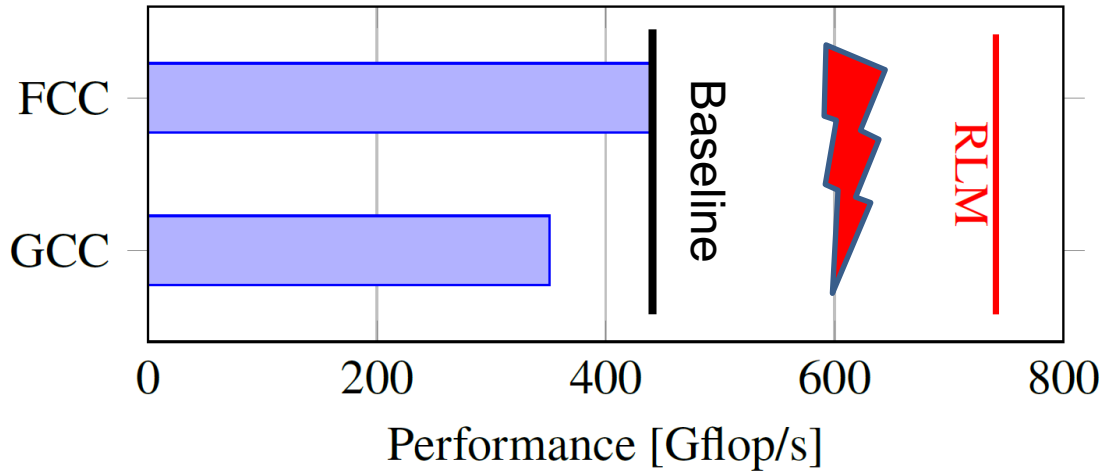
Observed performance

- Starting point: RIRI layout, **ACLE intrinsics**, GCC/FCC
- 1320 flops/LUP (theoretical)
- Measured code balance: 1500 byte/LUP

$$B_c \approx 1.14 \frac{\text{byte}}{\text{flop}}$$

- A64FX (FX1000): $B_m = 0.25 \frac{\text{byte}}{\text{flop}}$ → expect memory bound

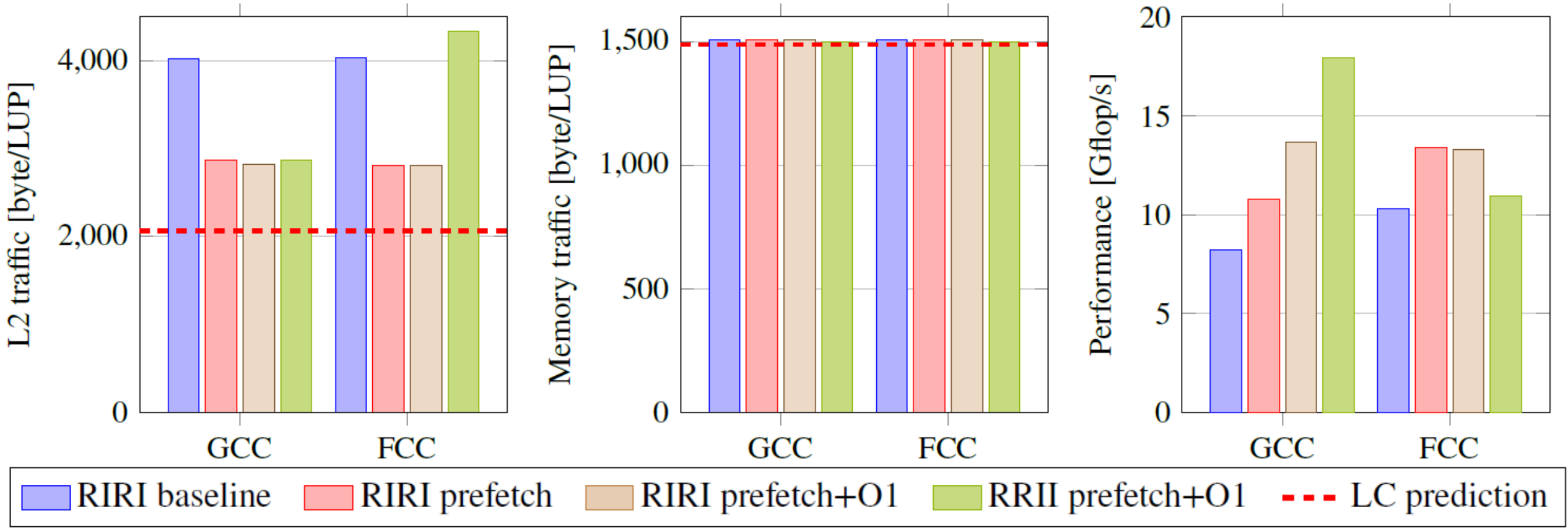




Summary of optimizations for DW

- Software prefetching decreases L2 data volume
- `-O1` makes compiler obey the ordering hints in the computational kernel (more efficient OoO execution)
- RRII data layout
  - Prevents use of complex arithmetic instructions **`fcmla`**/**`fcadd`**
  - Removes imbalance between FLA and FLB ports in the core
  - Some register spills occur, but still better than RIRI
  - Measurement falls short of ECM prediction by 2.3x (GCC) or 3.1x (FCC)

DW kernel optimizations and ECM model

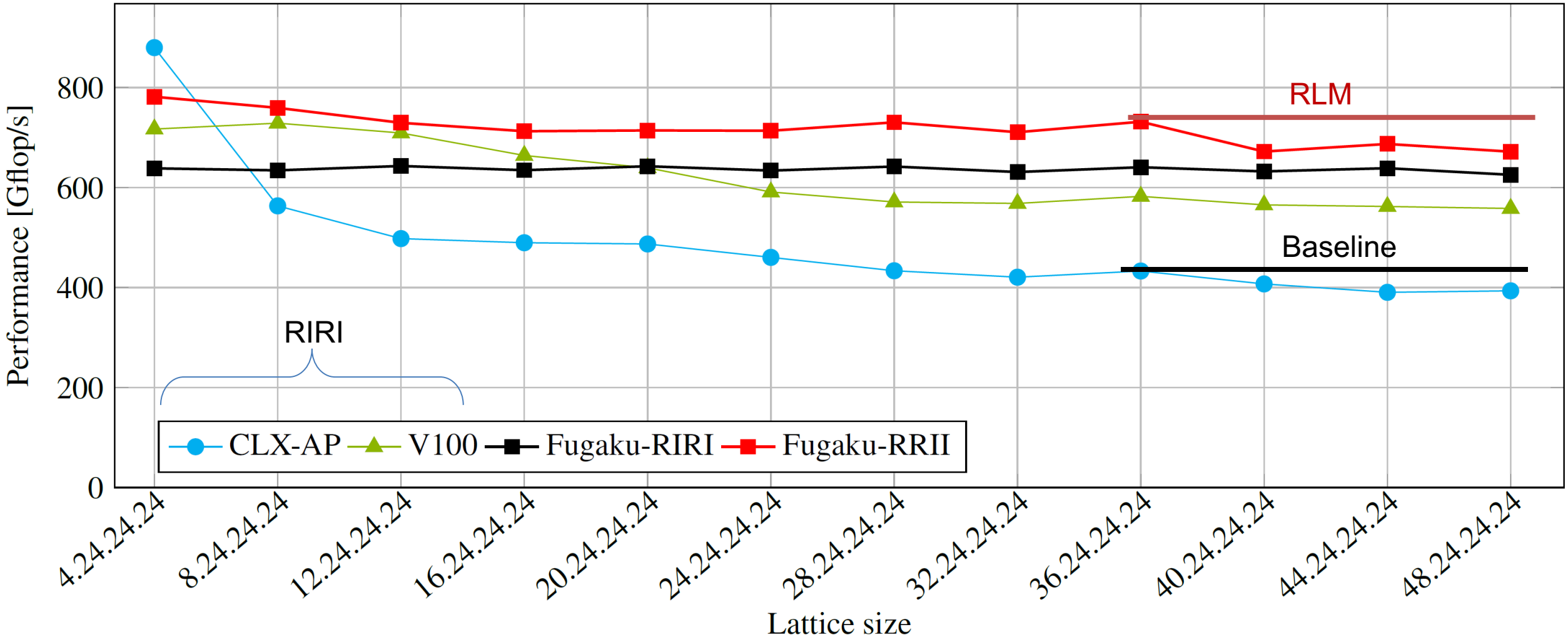


LIKWID TOOLS

available for A64FX

https://nhr.fau.de/research/tools/likwid/

Comparison with other architectures



Summary

- ECM model constructed for single-core performance of A64FX
- Partially overlapping memory hierarchy → high single-core memory bandwidth (even more so with large pages)
- If performance is bad, the single-core performance is usually the culprit
- SpMV requires proper data format for efficient single-core execution
- DW kernel benefits from prefetching and OoO improvements
- Performance modeling is invaluable for navigating optimization efforts

NHR FAU

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Thank You.

RIKEN
R-CCS
RIKEN
Center for
Computational Science

SFB
TRR55

KONWIHR

NHR