YaskSite: Stencil Optimization Techniques Applied to Explicit ODE Methods on Modern Architectures

Christie Alappat, Johannes Seiferth, Georg Hager,
Matthias Korch, Thomas Rauber, Gerhard Wellein

International Symposium on Code Generation and Optimization (CGO 2021)

Introduction

UNIVERSITÄT BAYREUTH

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Auto tuning

2x

System of ordinary differential equations → Parallel Iterated Runge-Kutta (PIRK) method → Implementation? → Solution

```
/* Implementation A */
// BARRIER
for (int k=0; k<m; ++k) {
  RHS
  // BARRIER
  LC
  // BARRIER
}
// BARRIER
App
// BARRIER
Up
```

Which implementation performs best?

```
/* Kernel RHS_lj */
for (int j*=0; j*<N; ++j*)
  for (int i=0; i < s; ++s)
    F[i][jx][jy][jz] = %RHS
```

More optimizations? (Tiling, vector folding, …)

```
/* Implementation B */
// BARRIER
for (int k=0; k<m; ++k) {
  RHS_LC
  // BARRIER
}
RHS_App_LC
```

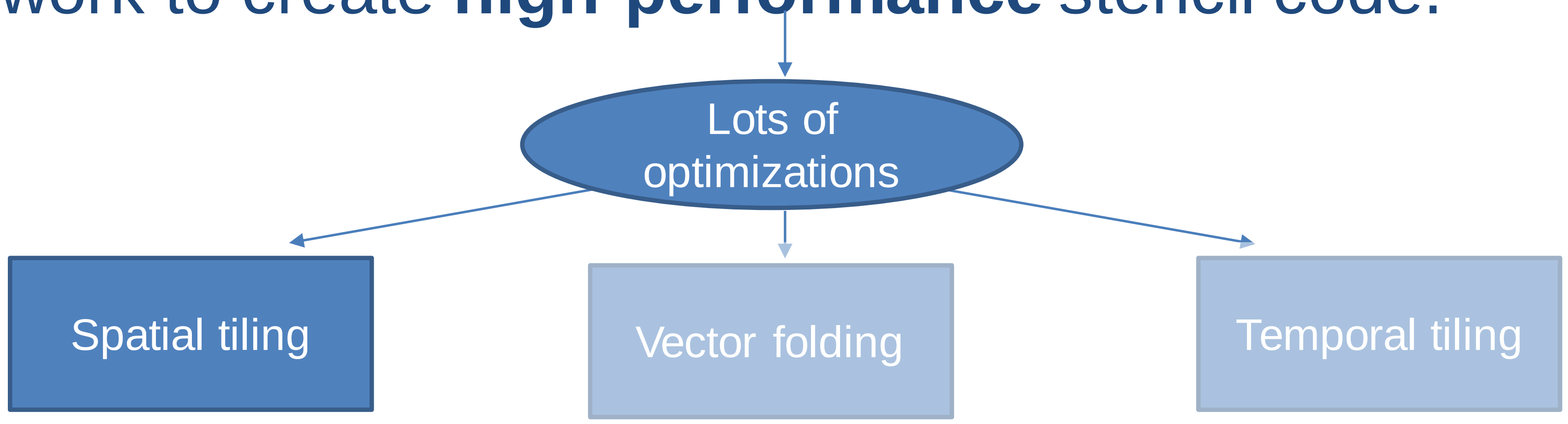
```
/* Kernel RHS_ij */
for (int i=0; i < s; ++s
  for (int j*=0; j*<N; ++j*)
    F[i][jx][jy][jz] = %RHS
```

YaskSite

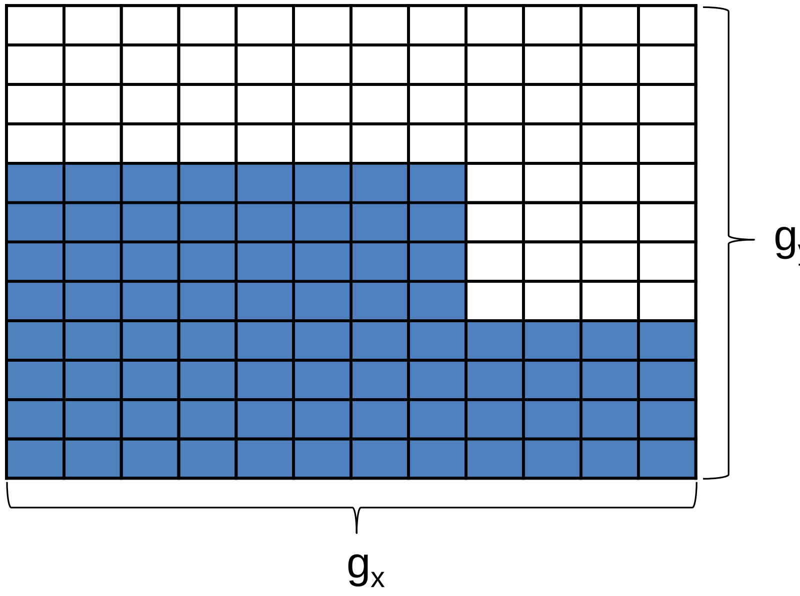
YASK & YaskSite

UNIVERSITÄT BAYREUTH | FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

- Framework to create **high-performance** stencil code.

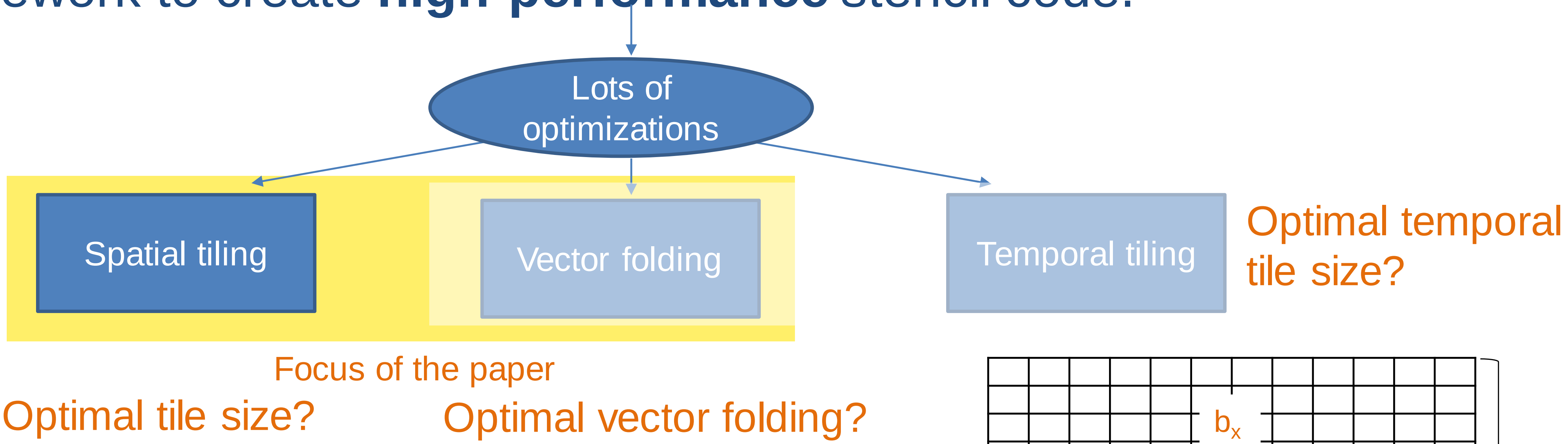



```
for(int t=1; t<g_t; ++t)
  #pragma omp parallel for collapse(3) schedule(static,1)
  for(int begin_b•=0; begin_b•<g•; begin_b•+=b•)
    for(int•= begin_b•;•< begin_b•+b•;•=•+ 1)
      out[t+1,x,y,z] = STENCIL(in[t, x, y, z])
```

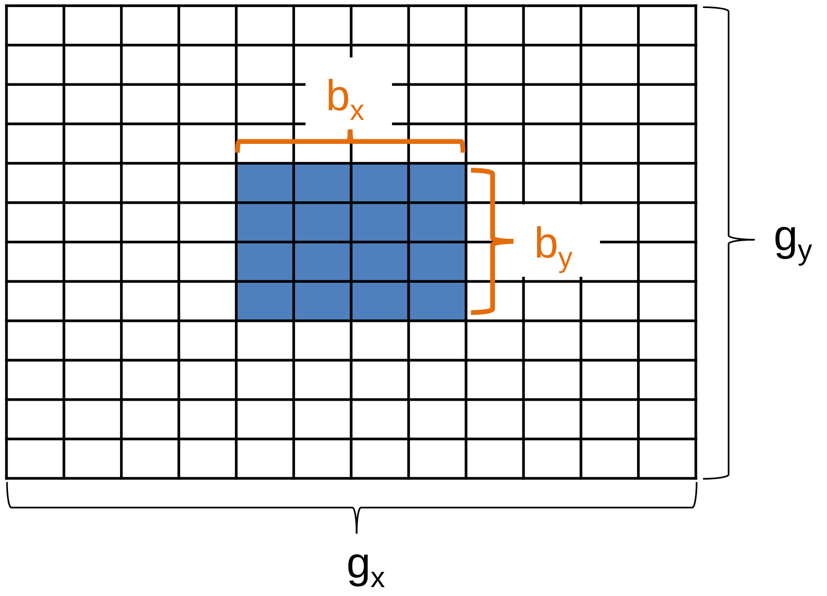


YASK - Yet Another Stencil Kit

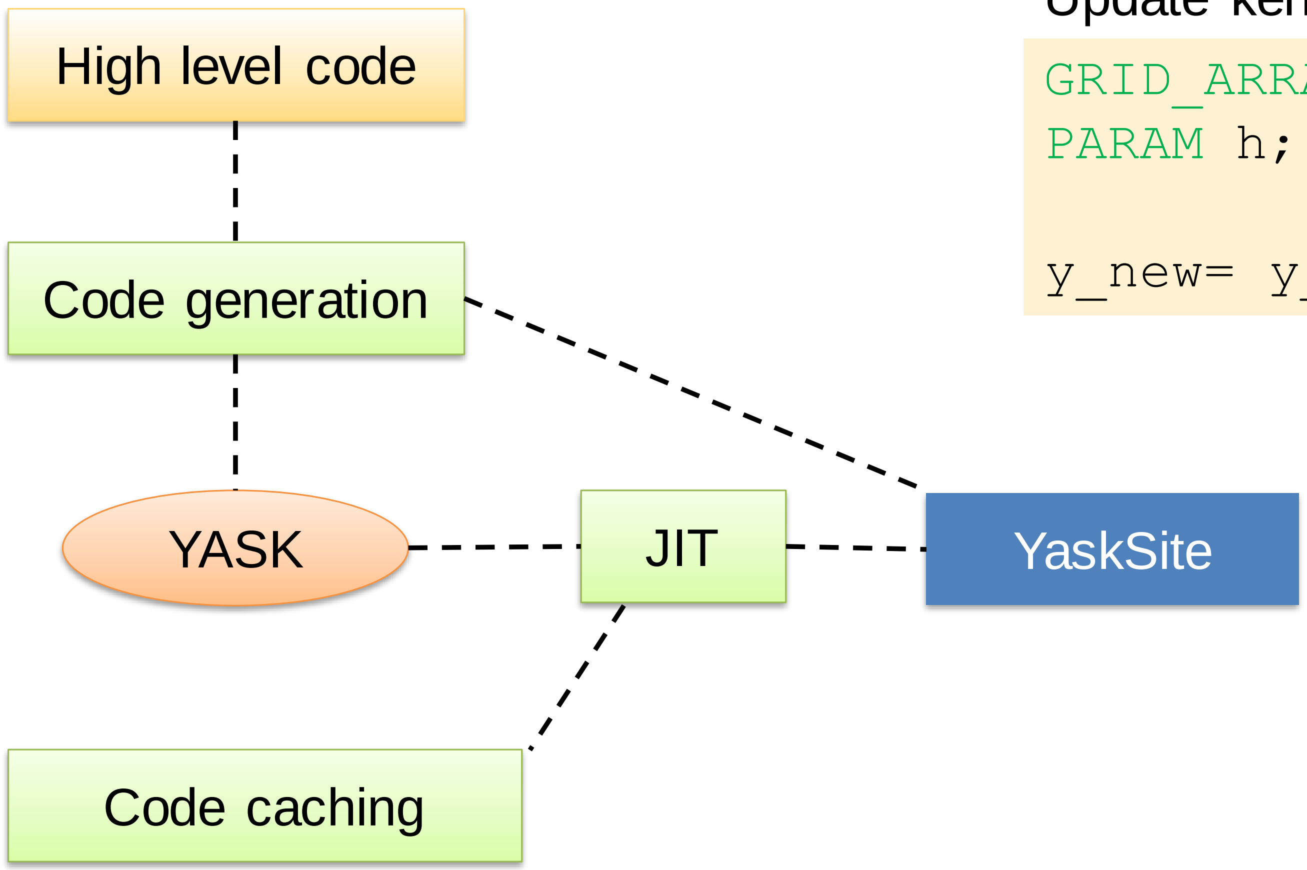
- Framework to create **high-performance** stencil code.




```
for(int t=1; t<g_t; ++t)
  #pragma omp parallel for collapse(3) schedule(static,1)
  for(int begin_b•=0; begin_b•<g•; begin_b•+=b•
    for(int•= begin_b•;•< begin_b•+b•;•=•+ 1)
      out[t+1,x,y,z] = STENCIL(in[t, x, y, z])
```



YASKSITE

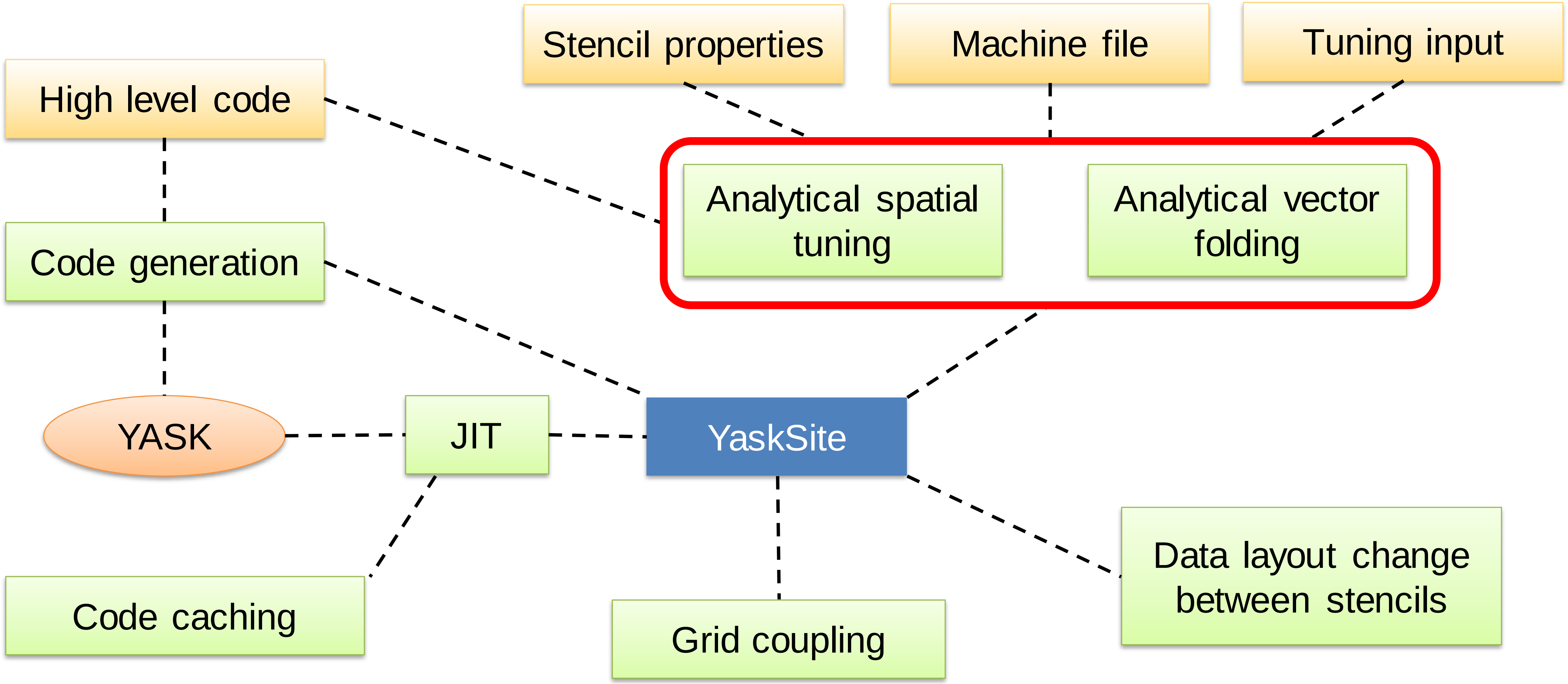


Update kernel

```
GRID_ARRAY y_new, y_old, dy;
PARAM h;

y_new= y_old + h * dy;
```

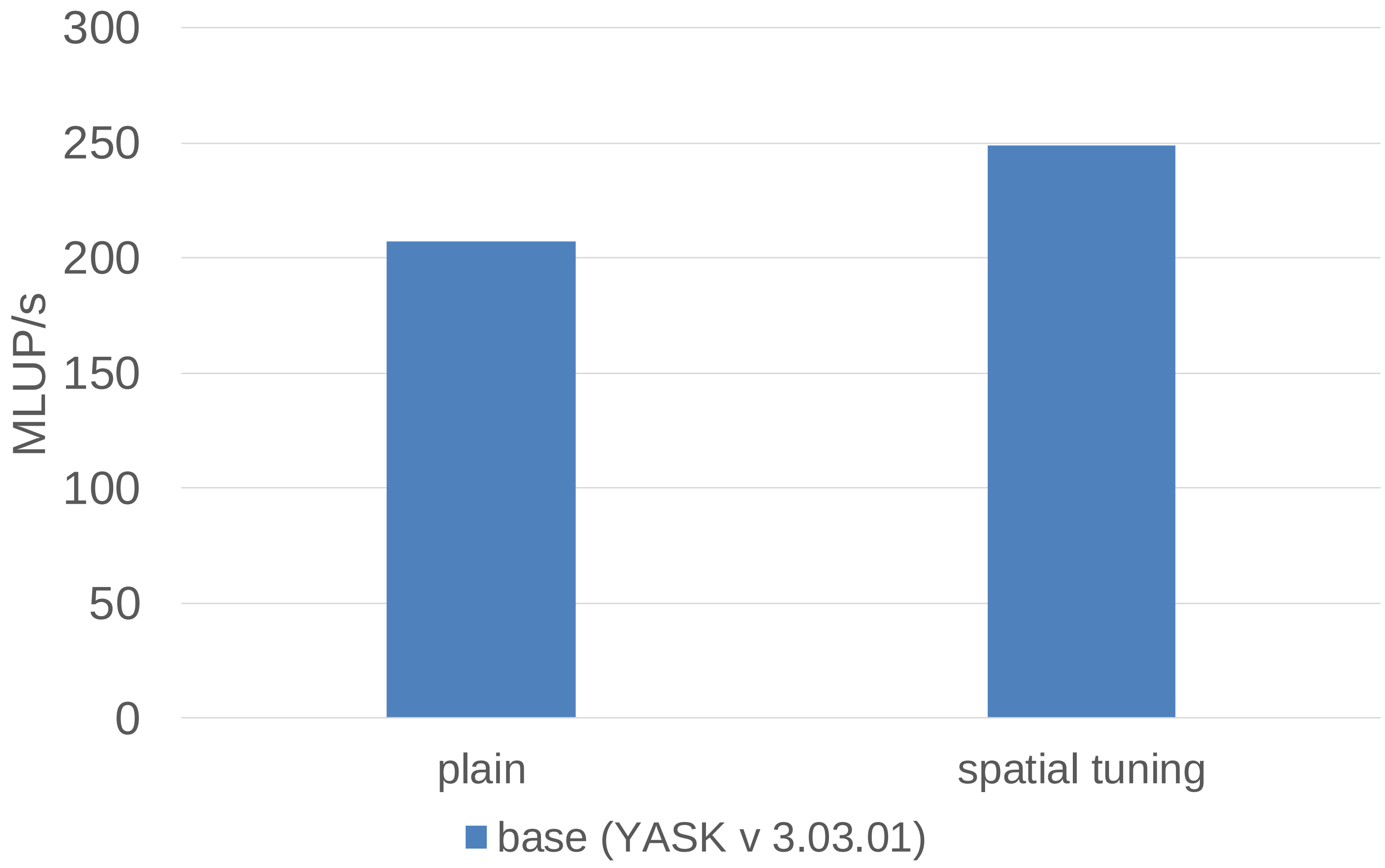
YASKSITE


High level code
Stencil properties
Machine file
Tuning input
Code generation
Analytical spatial tuning
Analytical vector folding
YASK
JIT
YaskSite
Code caching
Grid coupling
Data layout change between stencils

Performance modeling

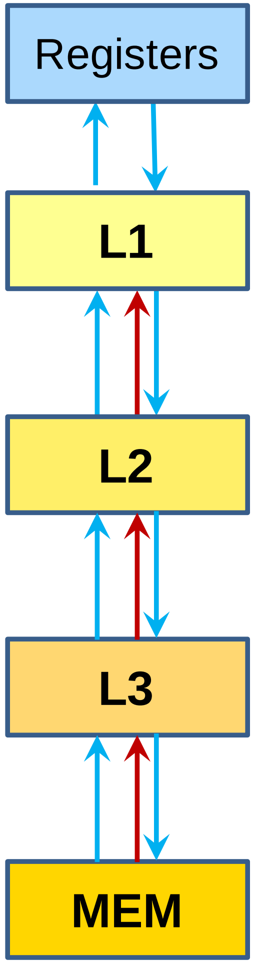
Motivation: Why modeling?

Performance of RHS_LC kernel on 1 socket (32 cores) of AMD ROME



Only 25% improvement from spatial tuning?

Execution-Cache-Memory* (ECM) performance model

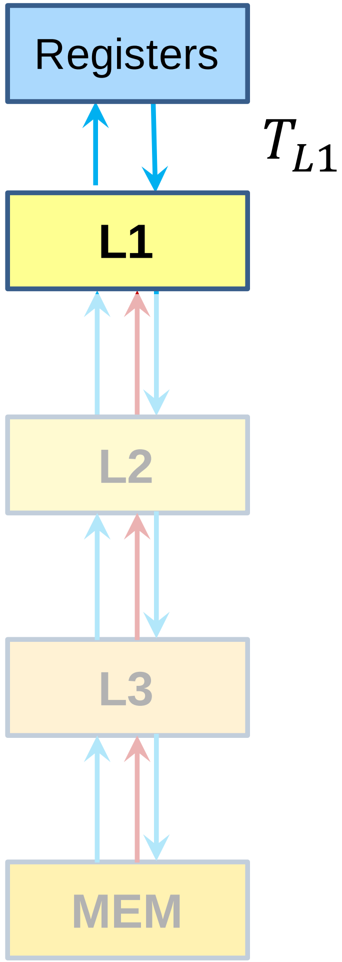


** Stengel et al., 2015. Quantifying Performance Bottlenecks of Stencil Computations Using the Execution-Cache-Memory Model.* https://doi.org/10.1145/2751205.2751240

UNIVERSITÄT BAYREUTH

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Execution-Cache-Memory (ECM) performance model

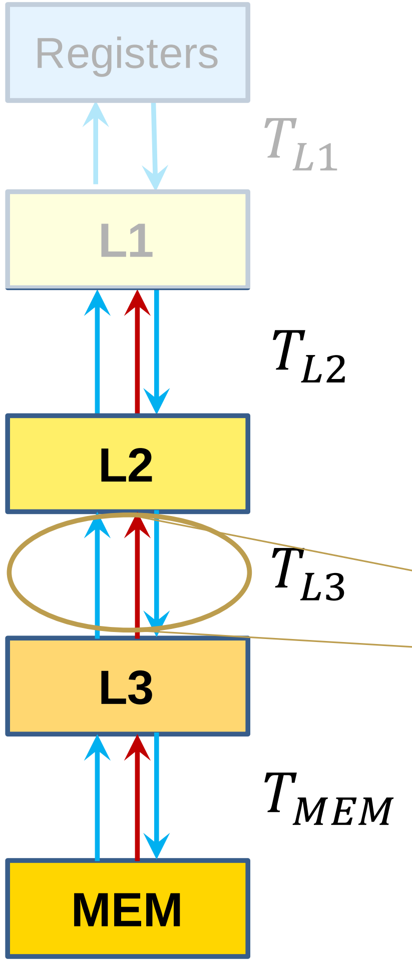


2 major components that influence performance:

1) In-core

Static code analysis tools:
IACA and OSACA

Execution-Cache-Memory (ECM) performance model

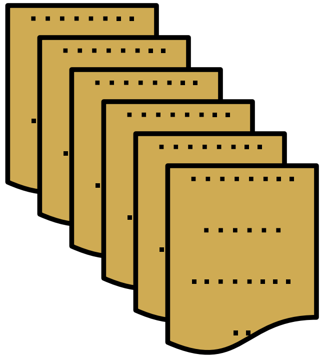


2 major components that influence performance:

1) In-core

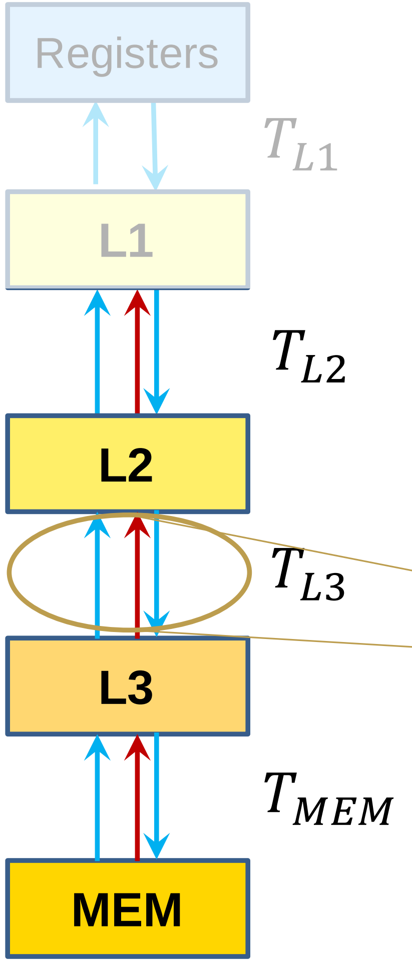
2) Data transfer through memory hierarchy

Amount of data



Analytical model

Execution-Cache-Memory (ECM) performance model

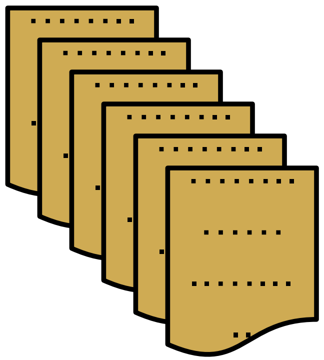


2 major components that influence performance:

1) In-core

2) Data transfer through memory hierarchy

Amount of data



L3

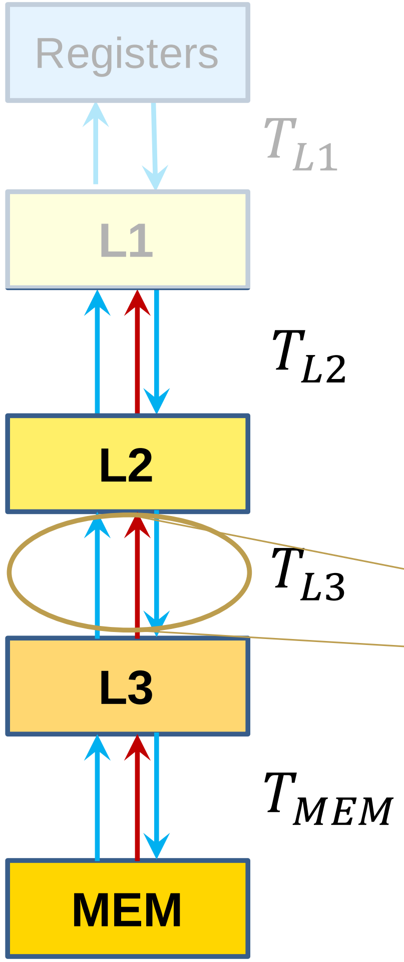
Rate of transfer

L2

UNIVERSITÄT BAYREUTH

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Execution-Cache-Memory (ECM) performance model



2 major components that influence performance:

1) In-core

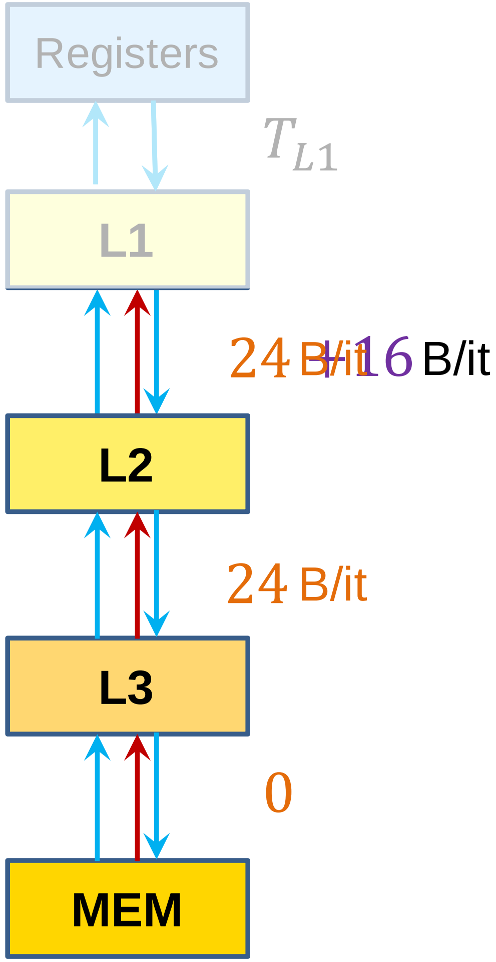
2) Data transfer through memory hierarchy

Depends on application

$$\frac{\text{Amount of data}}{\text{Rate of transfer}}$$

Depends on hardware

Amount of data : YASK application



Determine working-set size ➔ where data resides	2·8·N [B]	Assume fits in L3
Calculate minimum data traffic	8 + 16 [B/it]	
Account for stencil reuse	Layer condition analysis	Assume L1 cannot reuse

```
for(i =0; i<n; ++i) // n*n  = N
    for(j=0; j<n; ++j)
        b[i][j] = a[i-1][j] + a[i][j] + a[i+1][j];
```

Analytical model → Tuning

Layout condition analysis*

3D star stencil example with radius r

$$(N_s(2r+1)+(N_r-N_s)) * bz\,by\,d < CS$$

← If satisfied, reuse in x (outermost) dimension.

Tunable parameters

<u>Analytical spatial tuning</u>:
Solve for these parameters ($b_z\,, b_y$) to satisfy a certain cache (input) subject to constraints.

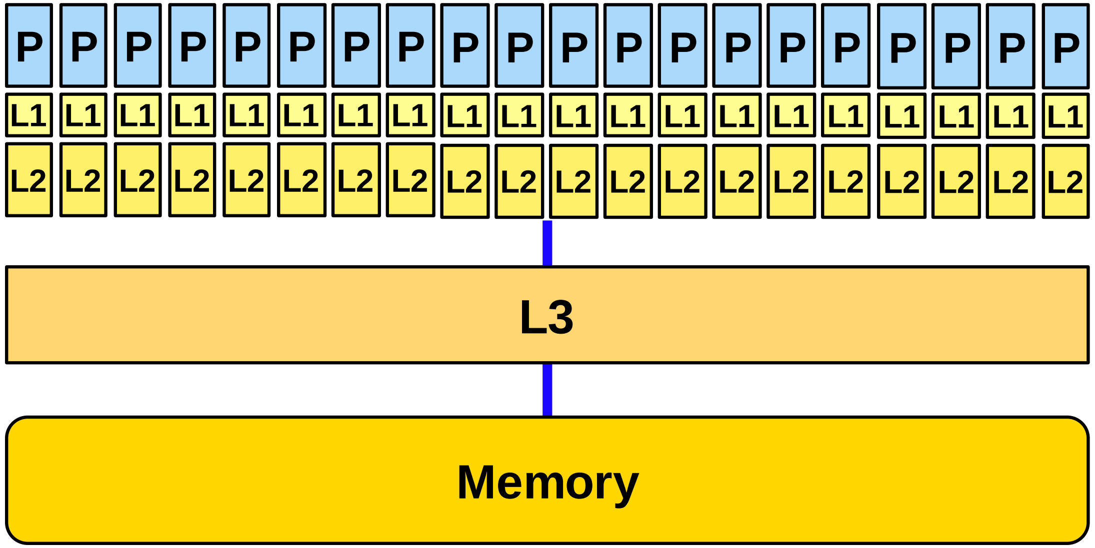
$$T = \max(T_{ol}, T_{L1} + T_{L2} + T_{L3} + T_{MEM}\downarrow)$$

ECM model prediction on CLX

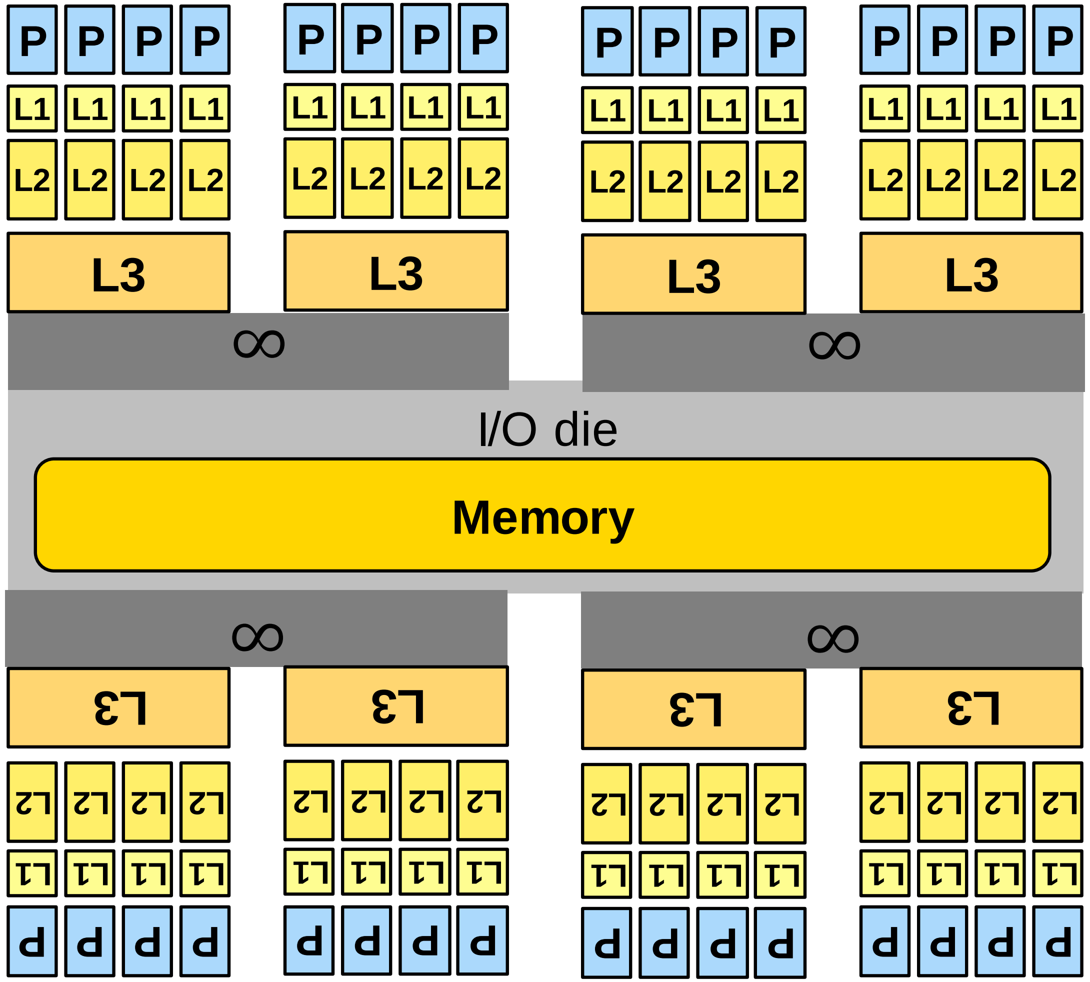
Gabriel Rivera and Chau-Wen Tseng. 2000. Tiling optimizations for 3D scientific computations. https://dl.acm.org/doi/10.5555/370049.370403

Test bed

Intel Xeon Gold 6248 (SNC-off mode)



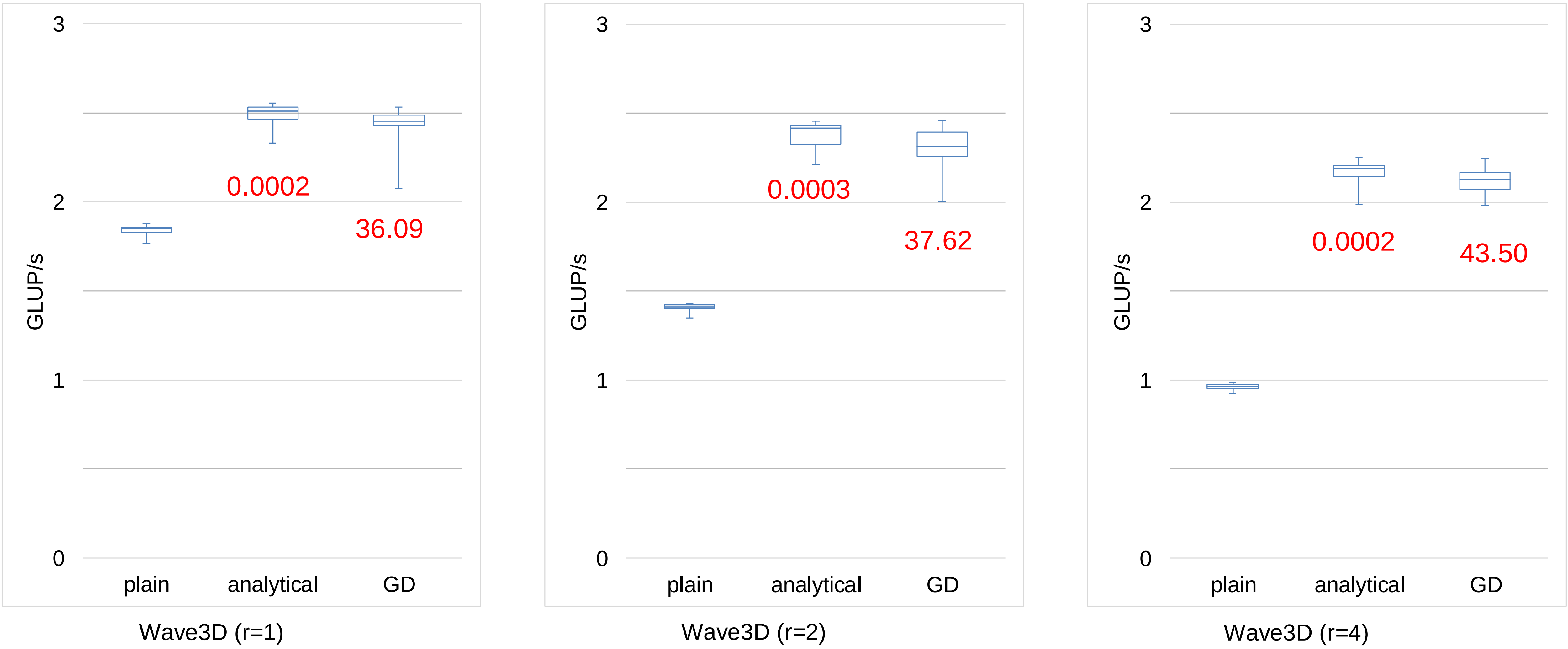
AMD EPYC 7452 – ROME (NPS1 mode)



Analytical tuning

Intel CLX

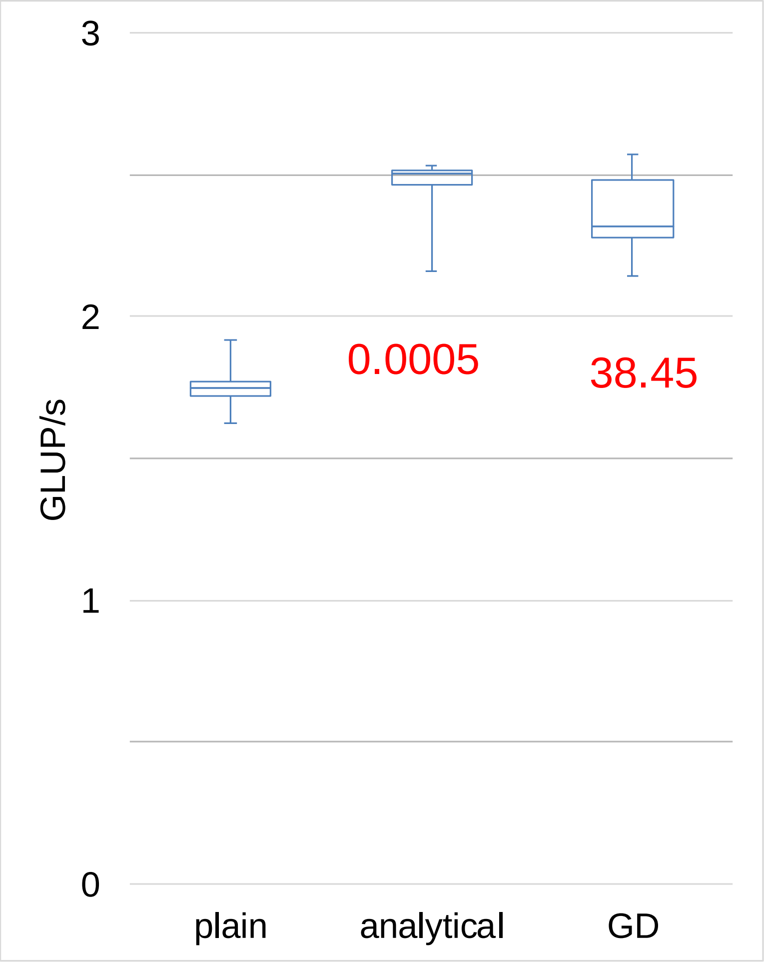
Tuning time in seconds



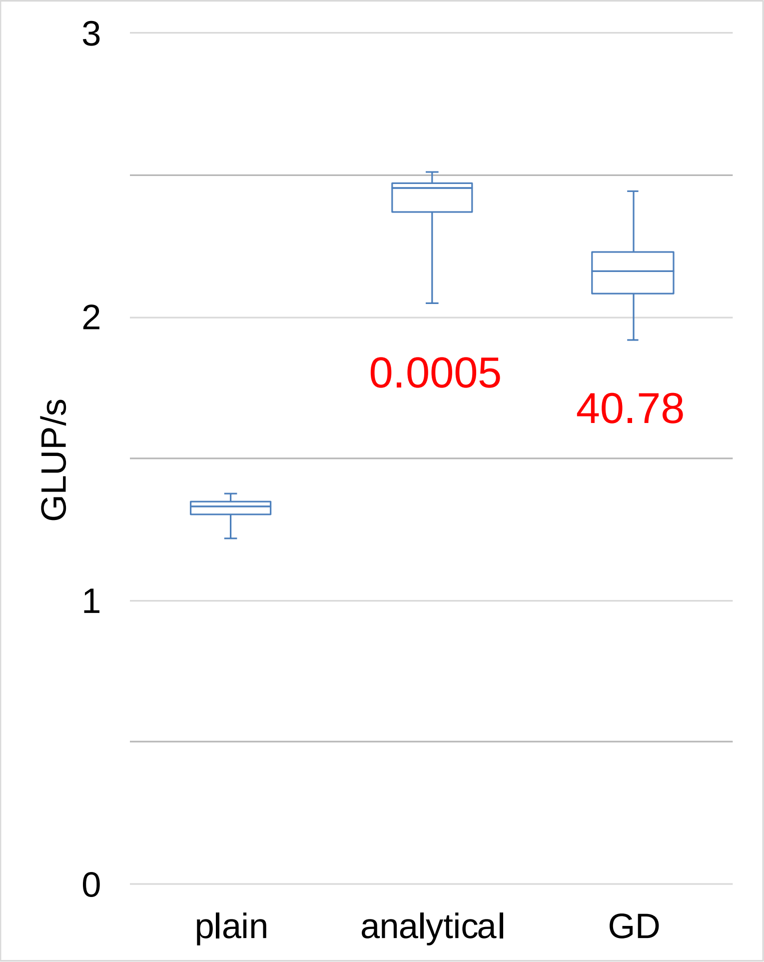
Analytical tuning

AMD ROME

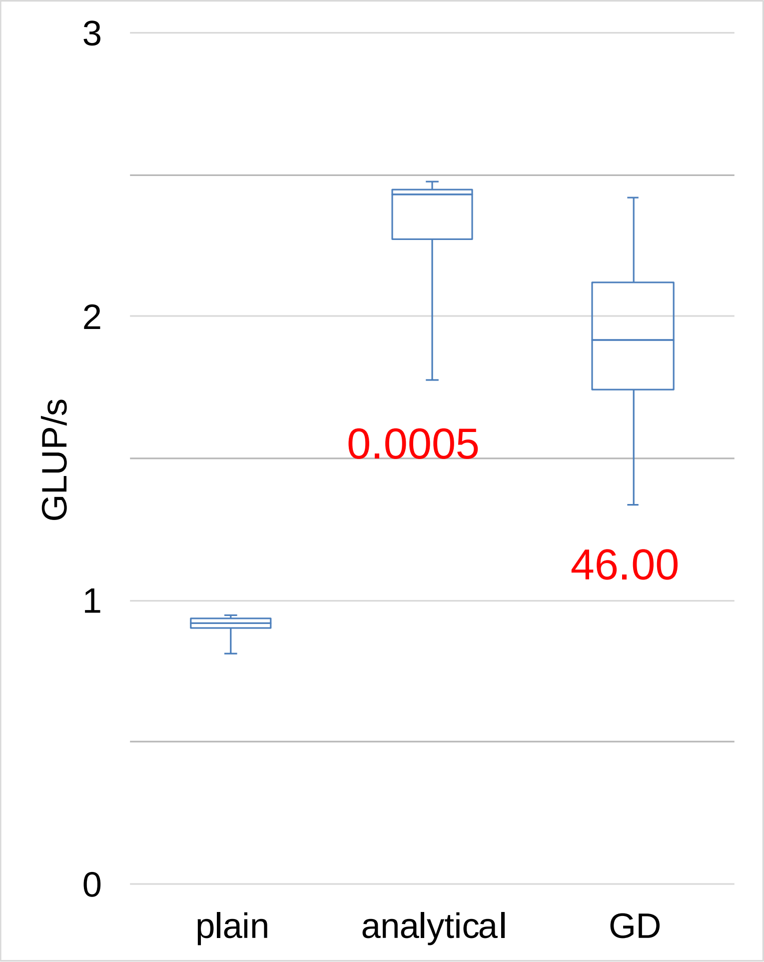
Tuning time in seconds



Wave3D (r=1)



Wave3D (r=2)



Wave3D (r=4)

Vector folding

Vector folding reduces traffic between L1 and registers.
The folding dimension is a parameter for vector folding.

Change in data layout

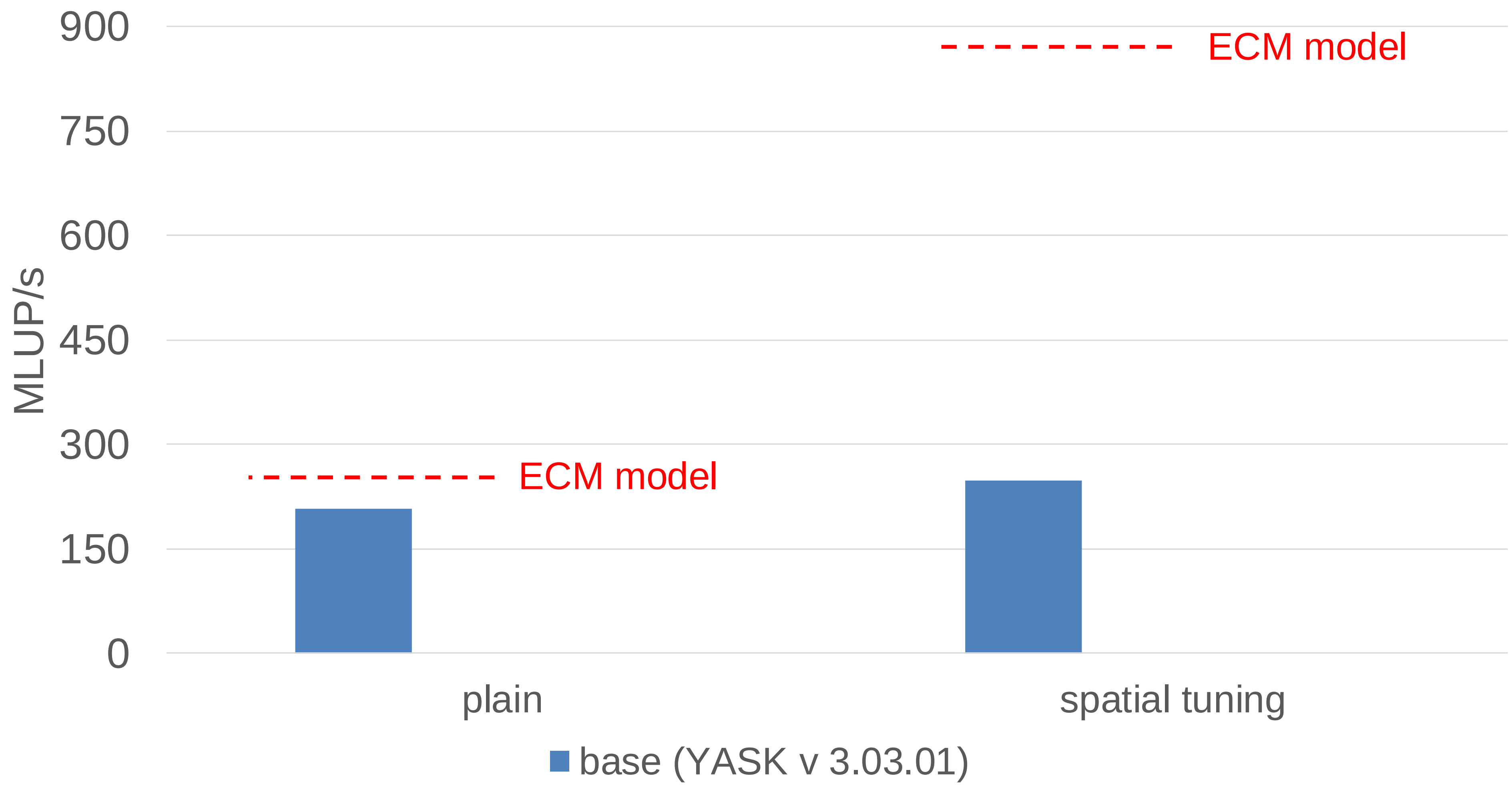
Change in layer condition analysis

Incorporate into performance model for analytical performance prediction and tuning

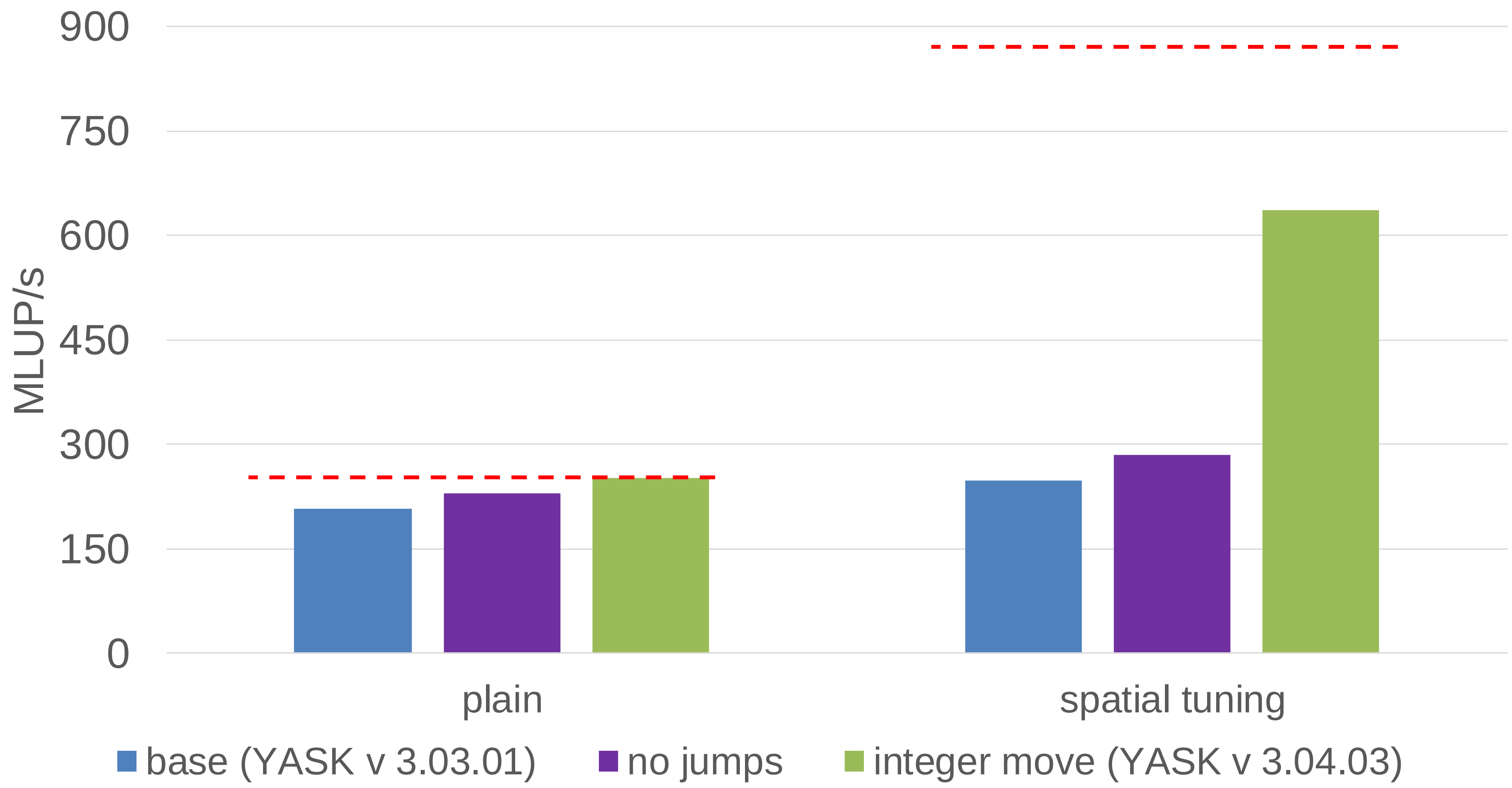
Yet another benefit of modeling

UNIVERSITÄT BAYREUTH
FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Performance of RHS_LC kernel on 1 socket (32 cores) of AMD ROME



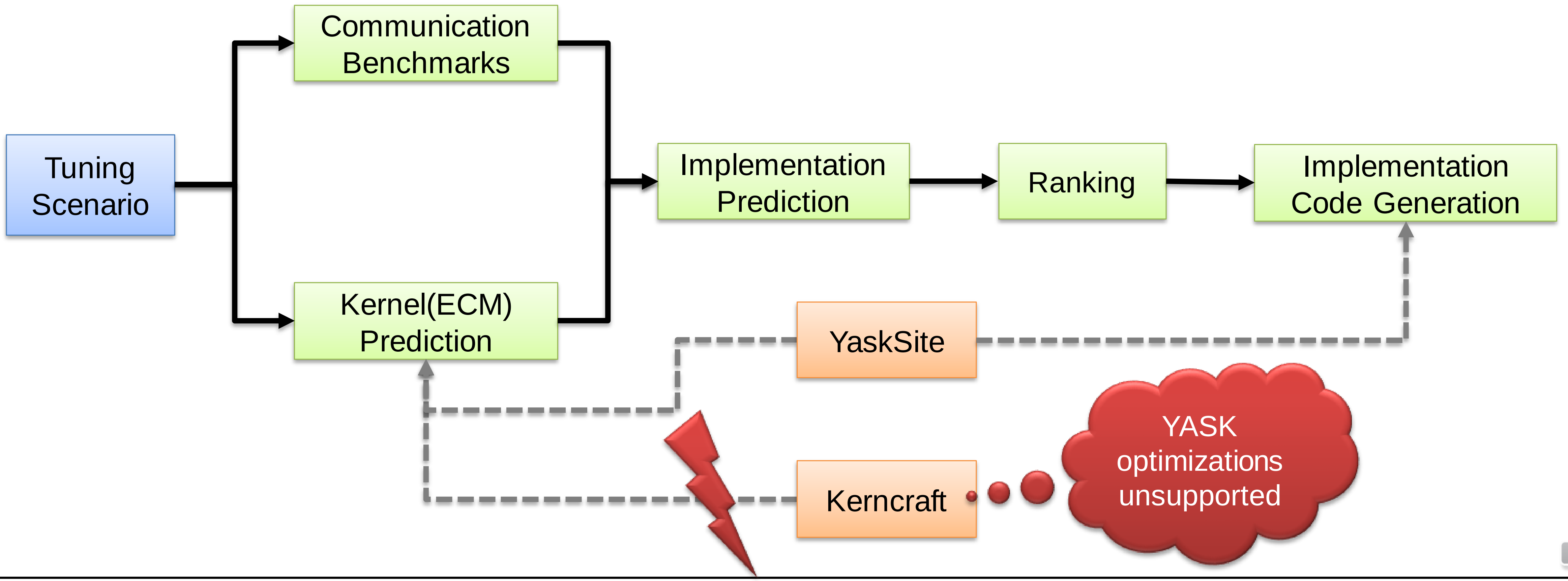
Performance of RHS_LC kernel on 1 socket (32 cores) of AMD ROME



Offsite: PIRK ranking

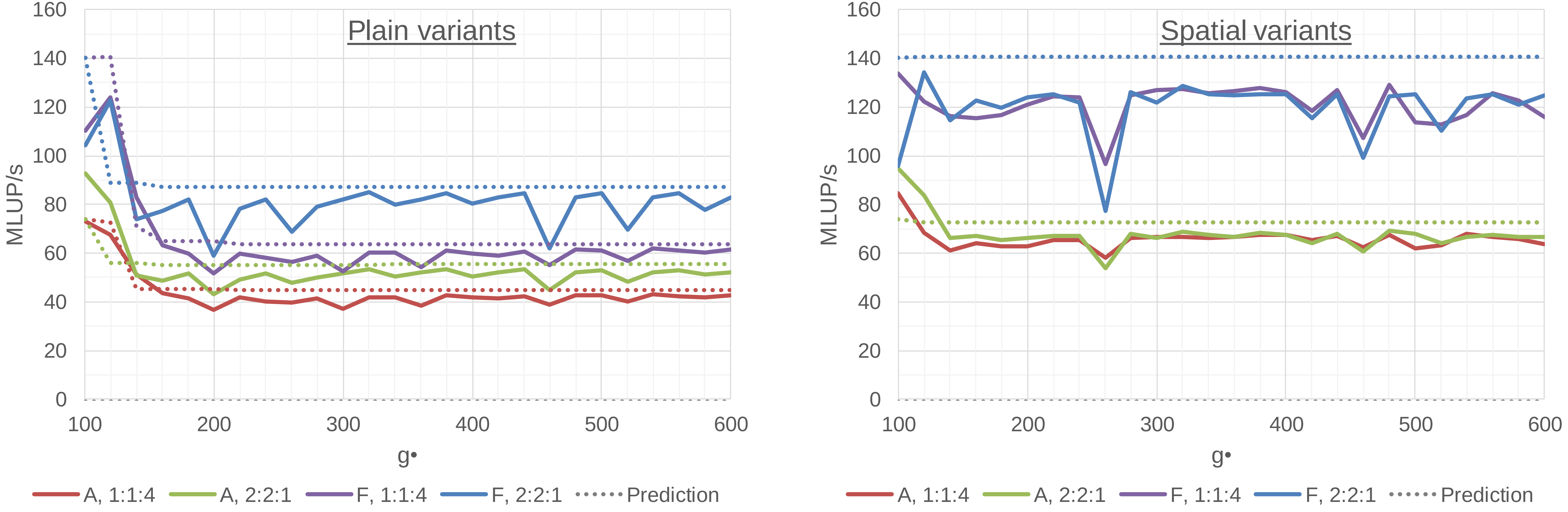
Offsite

- Tuning framework for explicit ODE methods
- YaskSite integrated as alternative ECM backend



Use-case: Tuning of PIRK variants

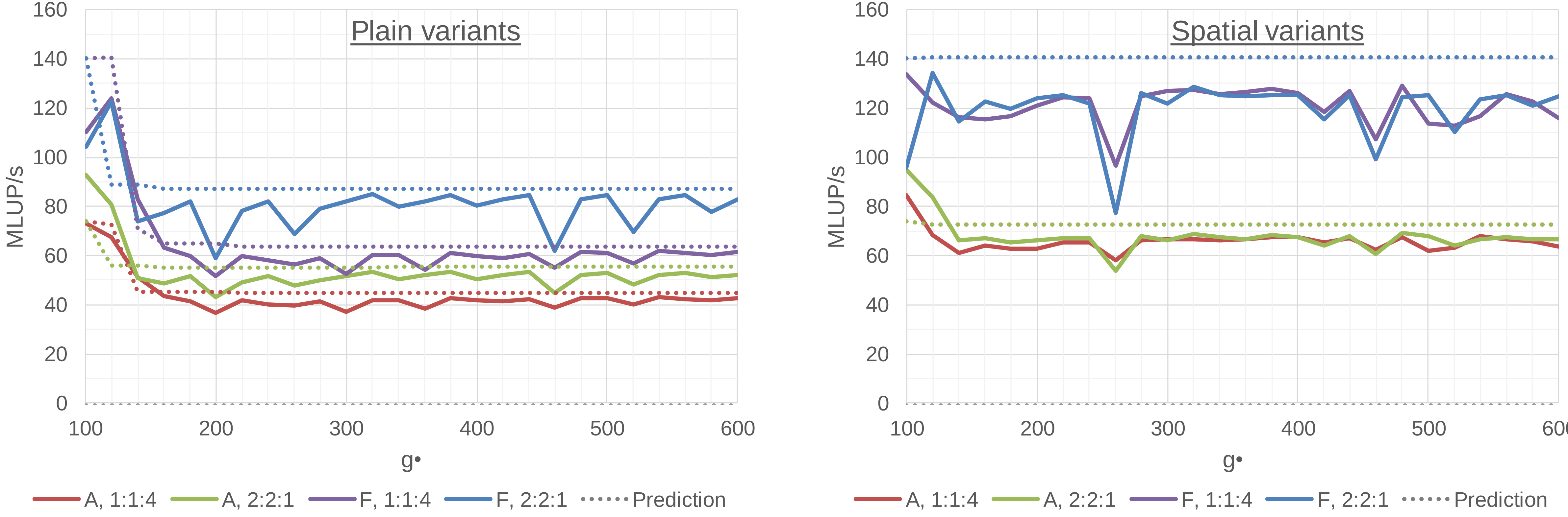
Predicted and measured performance of different PIRK variants for Wave3D (r=2) on AMD ROME



Use-case: Tuning of PIRK variants

UNIVERSITÄT BAYREUTH
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Predicted and measured performance of different PIRK variants for Wave3D (r=2) on AMD ROME



Offsite's ranking quality when using YaskSite as backend considering 16 different PIRK variants

Mean deviation (%)	*Maximum perf. loss*	*Mean perf. loss*
10.0 %	21.1 %	4.4 %

$$= \frac{best - selected}{selected} * 100$$

Conclusion

- YaskSite combines YASK with the analytical ECM model to successfully predict and tune the performance of stencil codes
  - ECM model’s layer condition analysis was extended to include vector folding and victim caches
  - First time application of the ECM model to AMD Rome
- Demonstrated YakSite’s usefulness to detect bottlenecks and guide performance optimization
- YaskSite can be integrated with external tools to tune more complex applications.

Thank you for your interest!

For more information please contact:
Christie Alappat Christie.alappat@fau.de, Johannes Seiferth Johannes.seiferth@uni-bayreuth.de

This work is supported by the German Ministry of Science and Education (BMBF) under the project SeASiTe.

https://github.com/seasite-project