Optimized Lattice Boltzmann Method for Fluid Simulation

Team 54:
Johannes Gasser, Rachel Schuchert
Severin Klapproth, Douglas Orsini-Rosenberg

Supervisor: Mikhail Khalilov

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lattice Boltzmann Method

Algorithm 1: LBM Simulation

Result: updated state S of fluid after T timesteps
initialize(S);
for $i \leftarrow 1$ **to** T **do**
  collision(S); `// adjust particle velocities based on local interactions`
  momentum(S) ; `// Update macroscopic properties from particle distributions`
  stream(S); `// Move particles to neighboring lattice nodes`
end

- **used for Fluid Simulation on a macroscopic scale**
- **We optimized all steps of the algorithm including 3 different boundary conditions for the stream step**
  - **Periodic**
  - **Lees Edwards**
  - **Couette**
- **Baseline Asymptotic Runtime: O(n^3)**

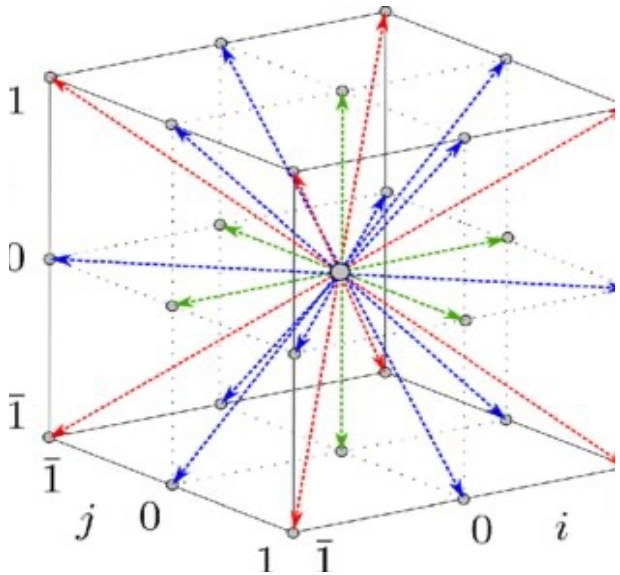


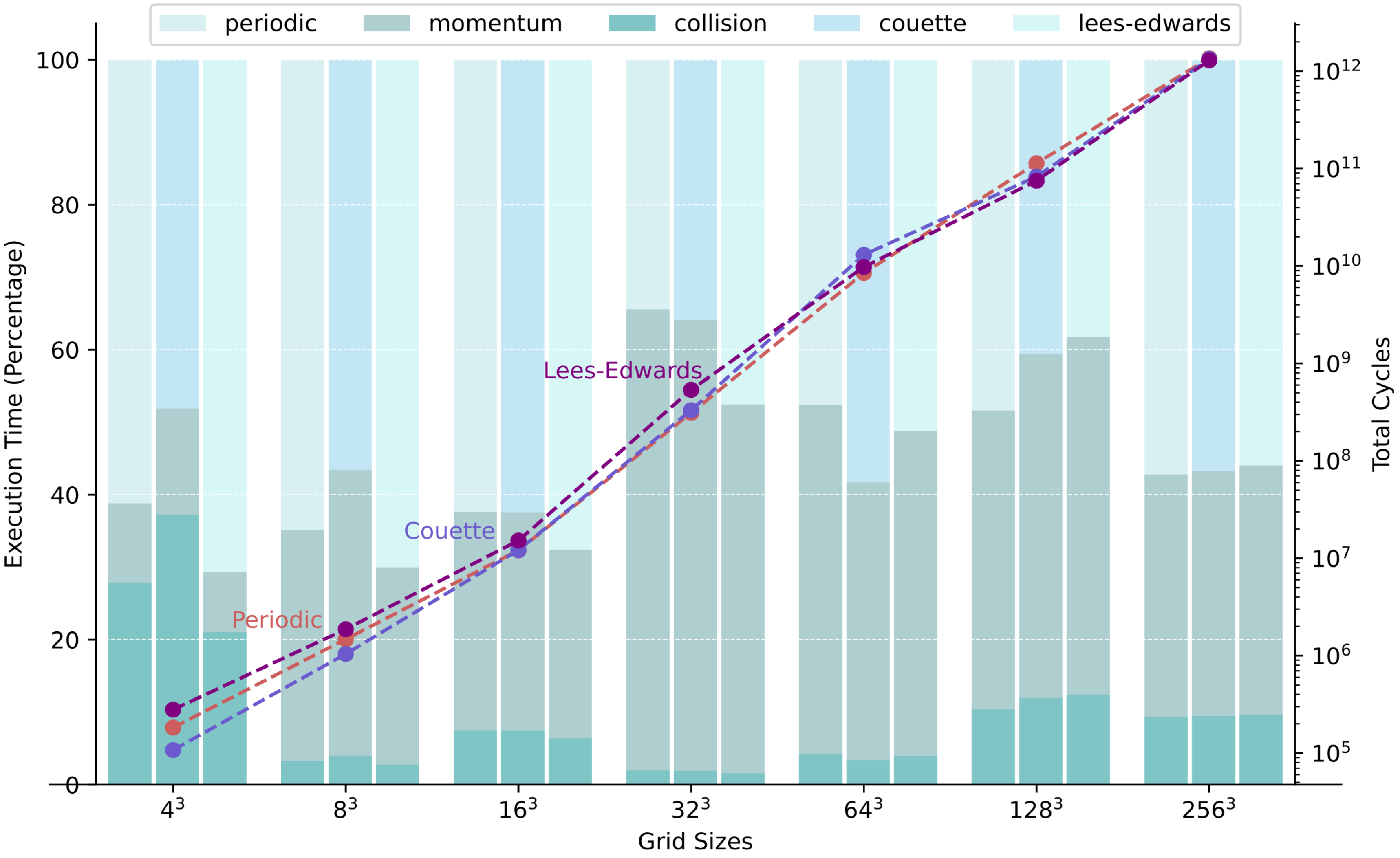
Fig 1.: Model of a D3Q27 Lattice

Infrastructure and Experimental Setup

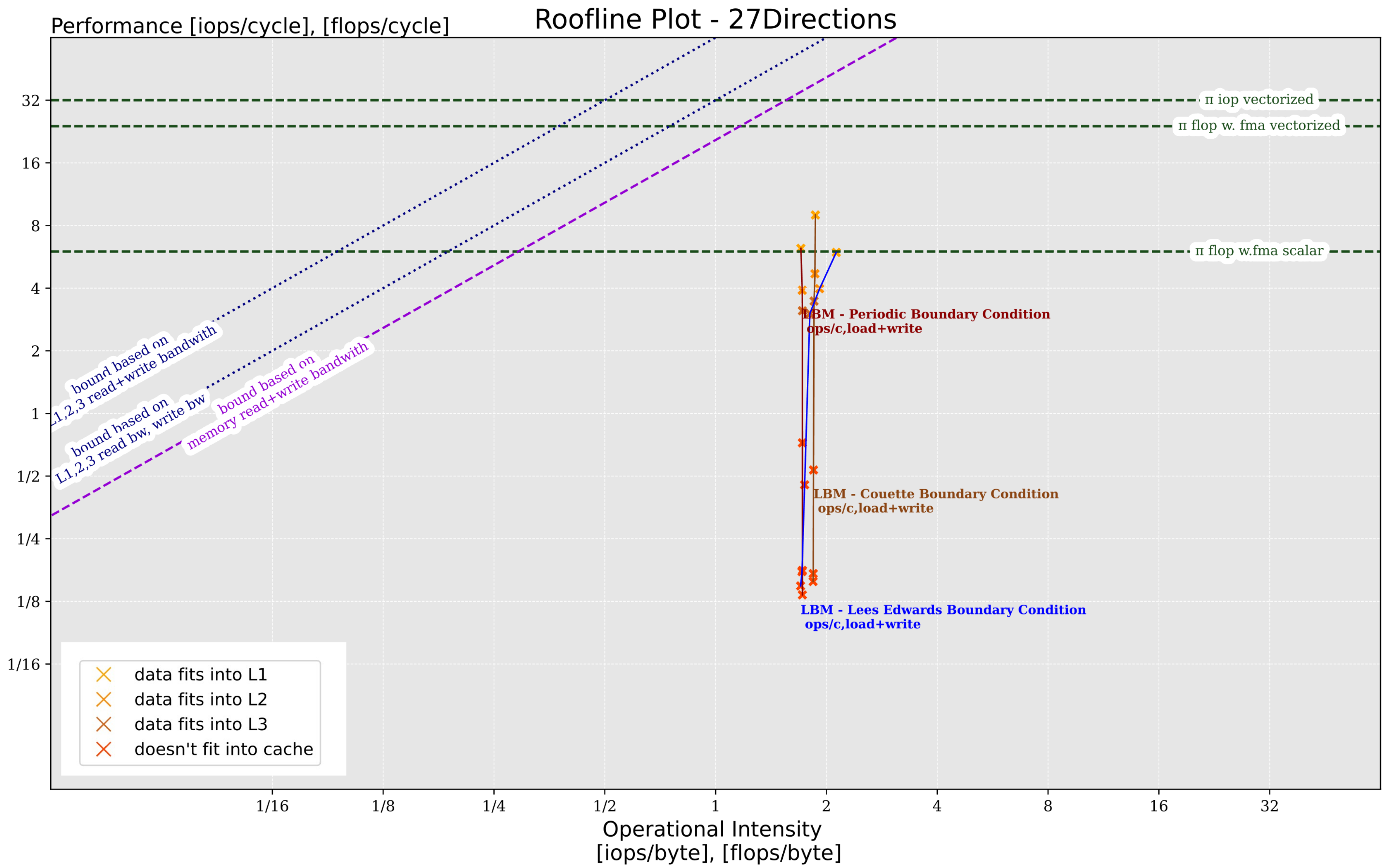
- **Automated testing and timing**
- **Cost Measure:**
  - $\boldsymbol{C(n,q) = N_{intops} * C_{intop} + N_{flops} * C_{flop}}$
    - *Convention:* $\boldsymbol{C_{intop} = C_{flop} = 1}$
- **Our experimental setup**
  - **Platform: Zen 3+ Architecture (***Processor: AMD Ryzen 7 PRO 6850U Specs: 2.7 GHz (L1: 512 KB, L2: 4 MB, L3: 16 MB)***)**
  - **Compiler and Flags:** *gcc -O3 -mavx2 -mfma -funroll-loops -march=znver3*

Baseline Implementation

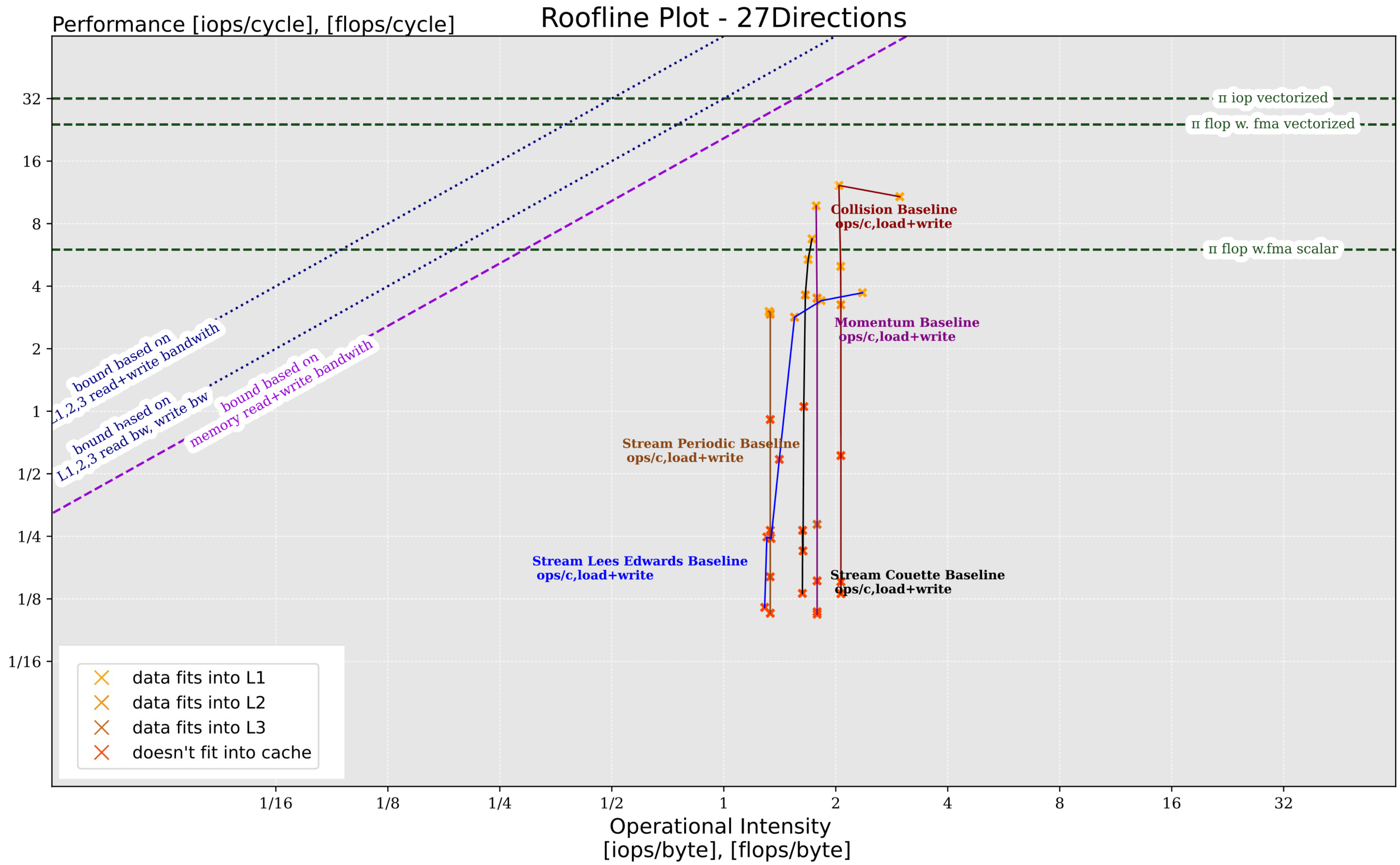
Boundary Condition Comparison - Bottleneck and Runtime Baseline



Roofline Baseline - D3Q27



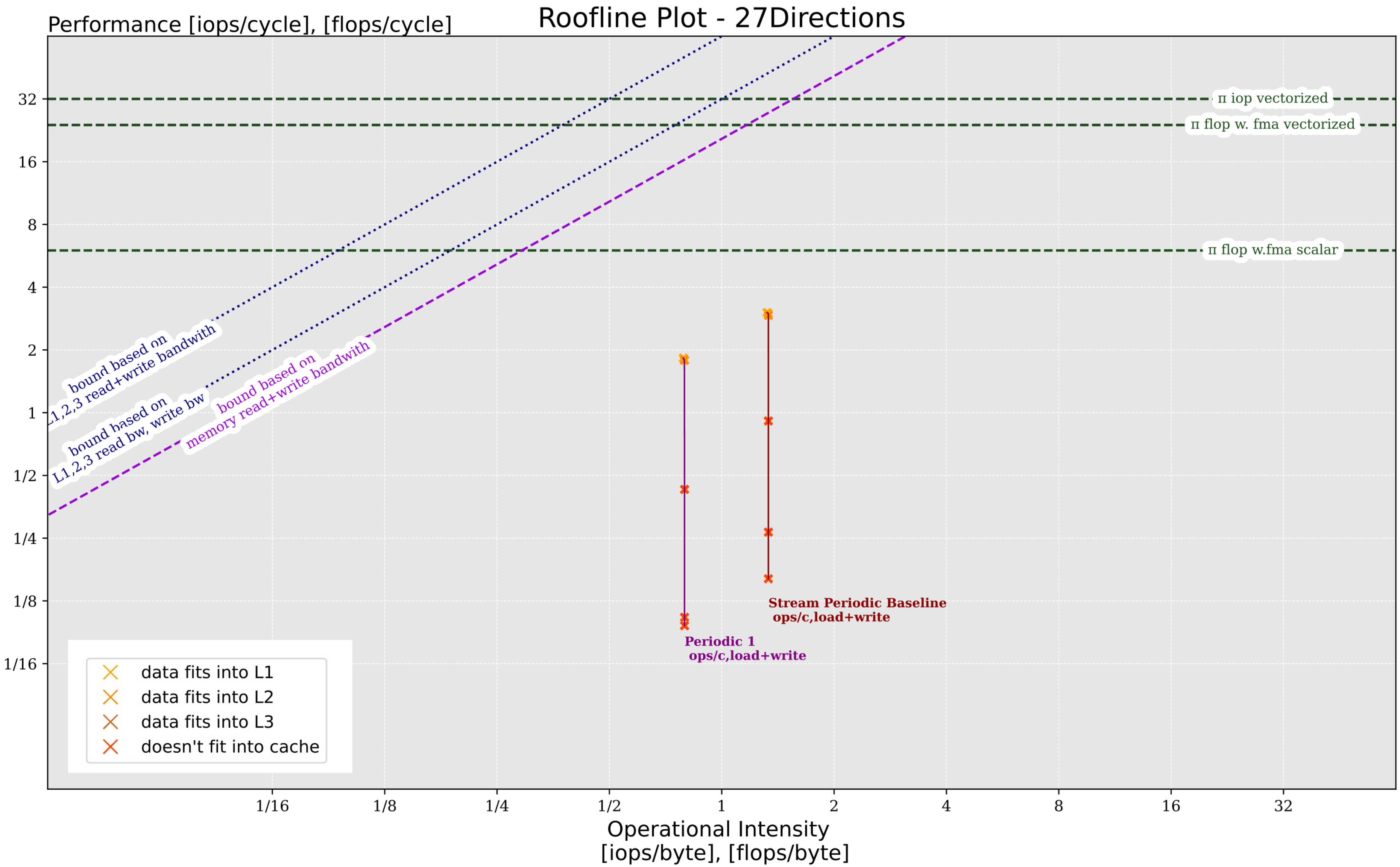
Roofline Baseline of each Individual Step



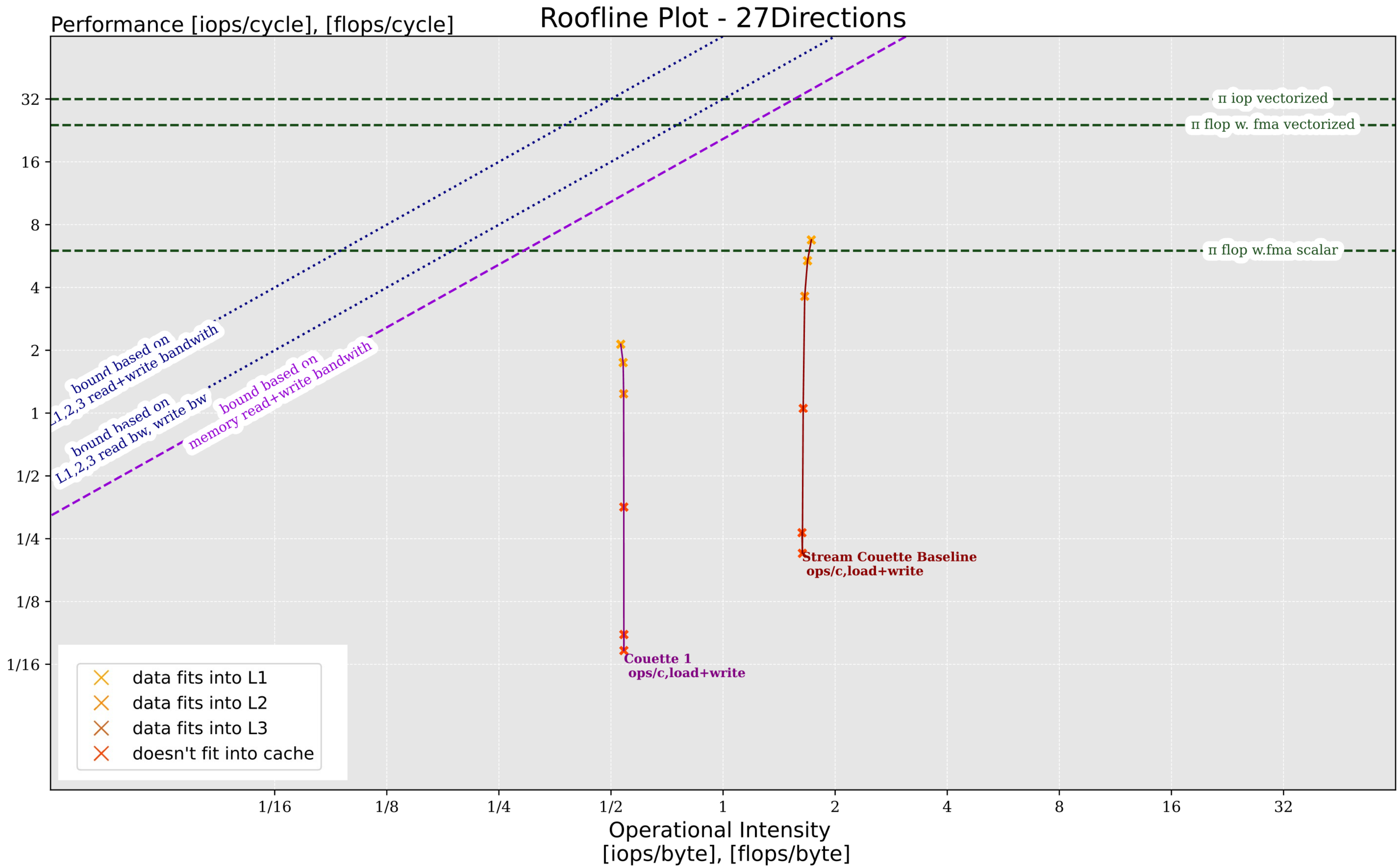
Optimization 1

Strength Reduction, Precomputations Loop Invariants, Function Inline Expansion, etc.

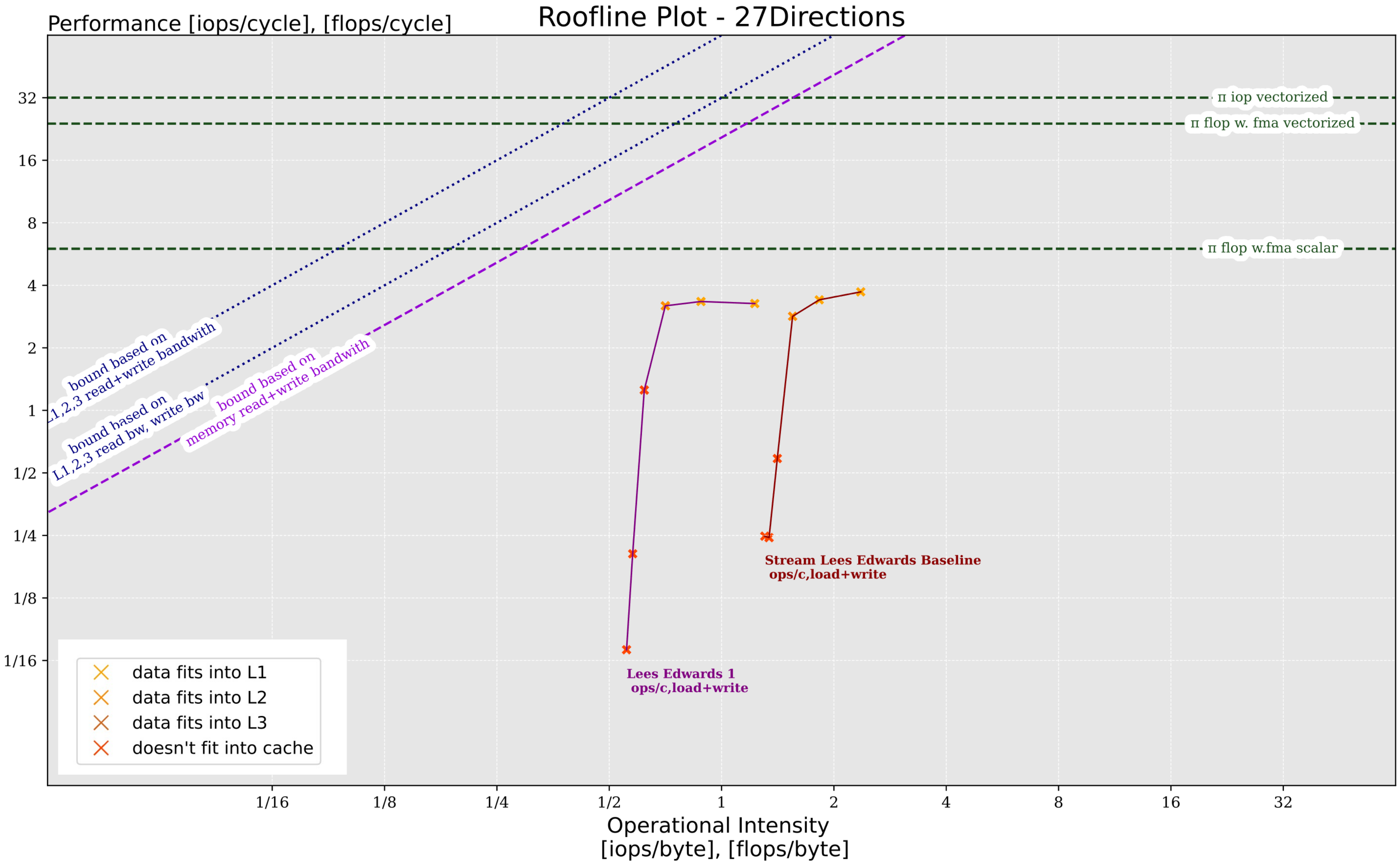
Optimization 1 – Periodic BC



Optimization 1 – Couette BC



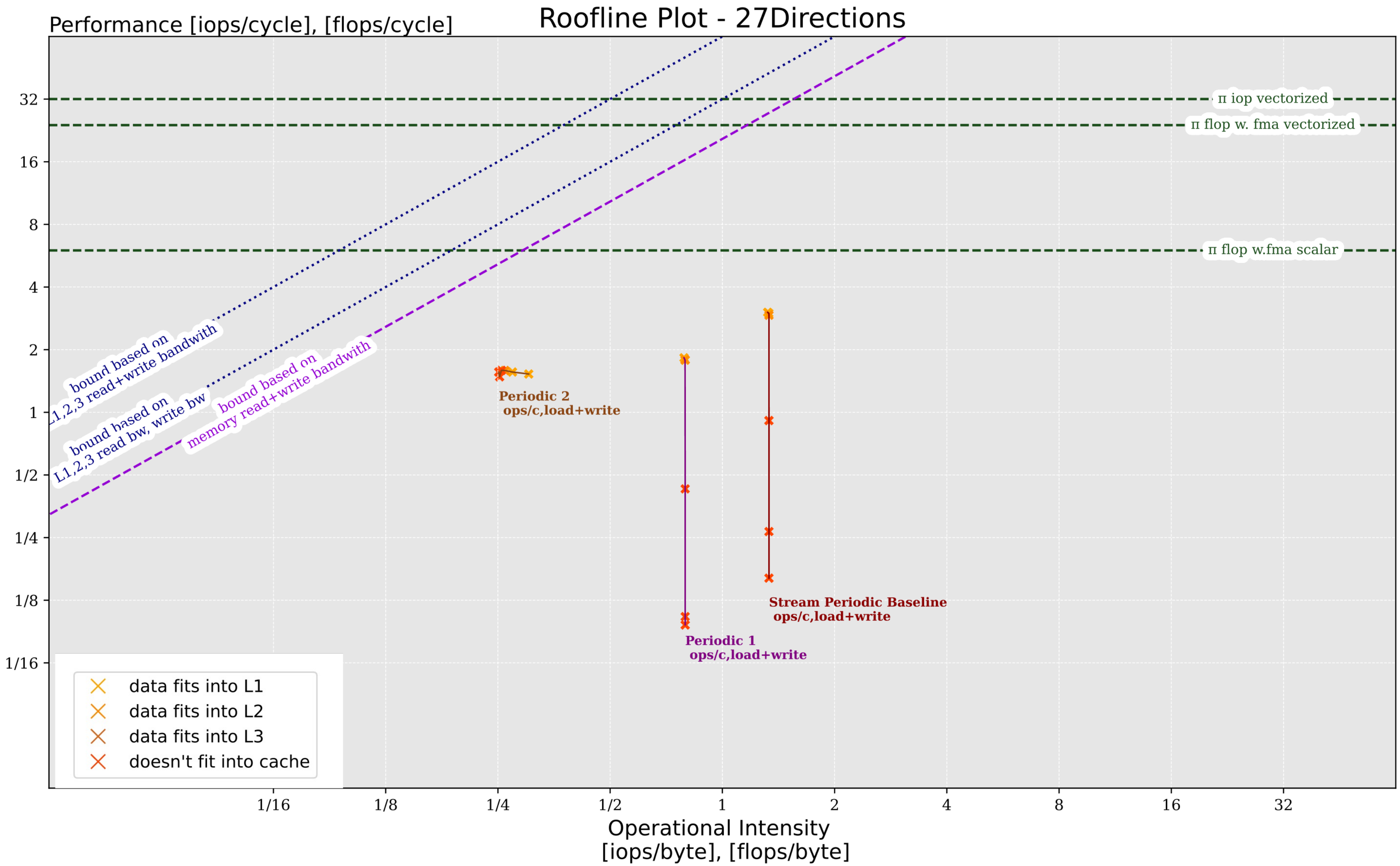
Optimization 1 – Lees Edwards BC



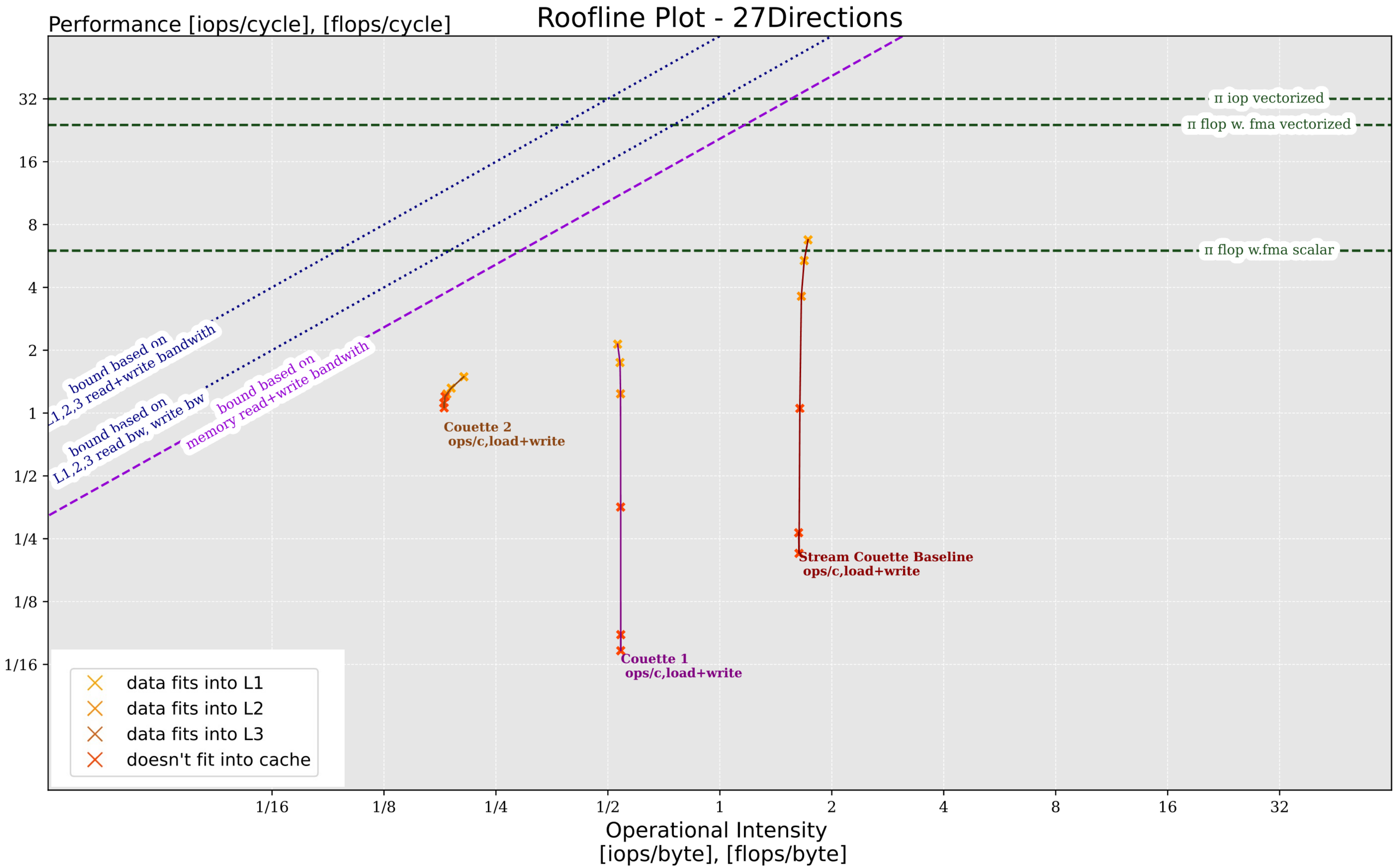
Optimization 2

Changing Loop Order, Taking into account Locality, Blocking, etc.

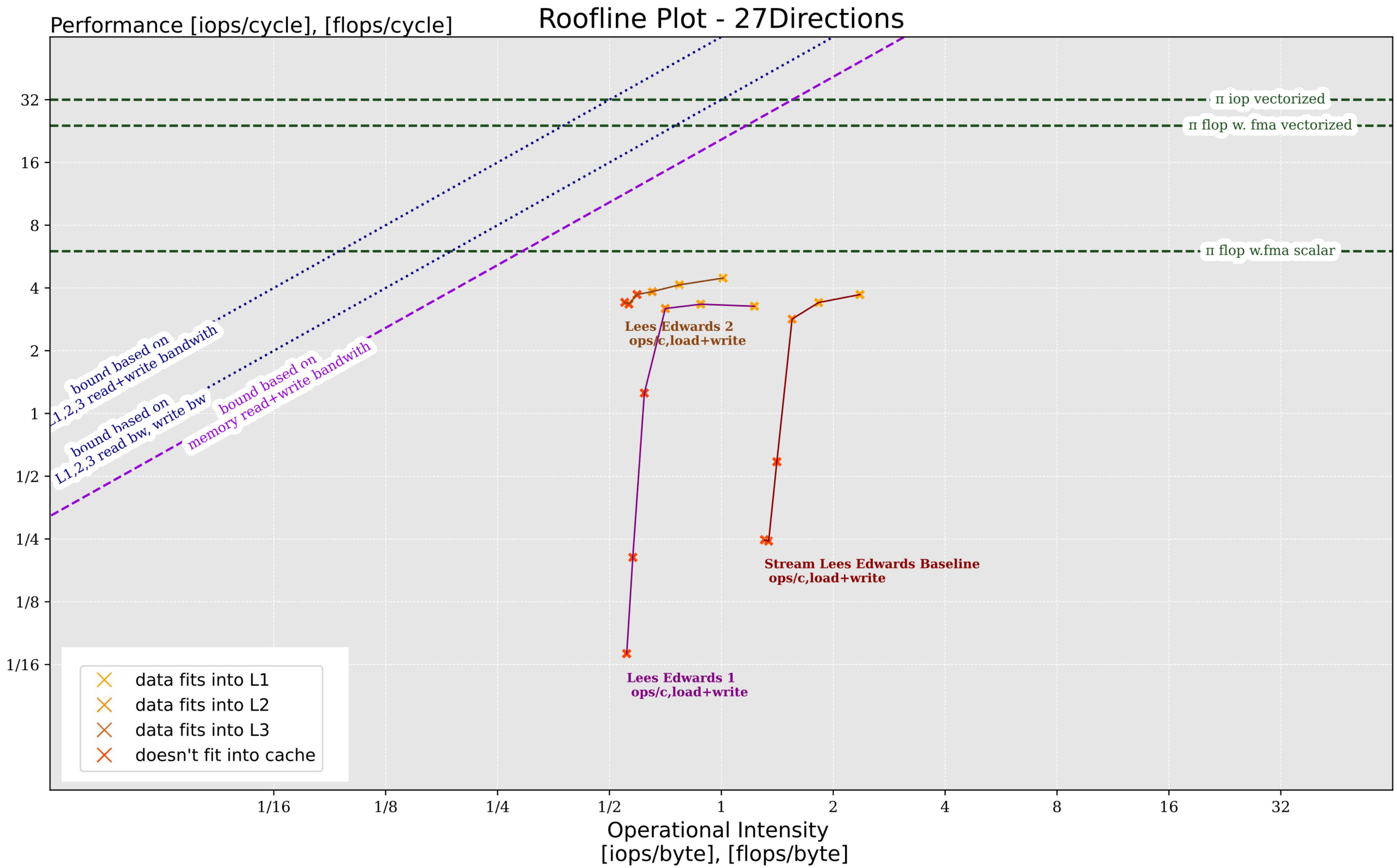
Optimization 2 – Periodic BC



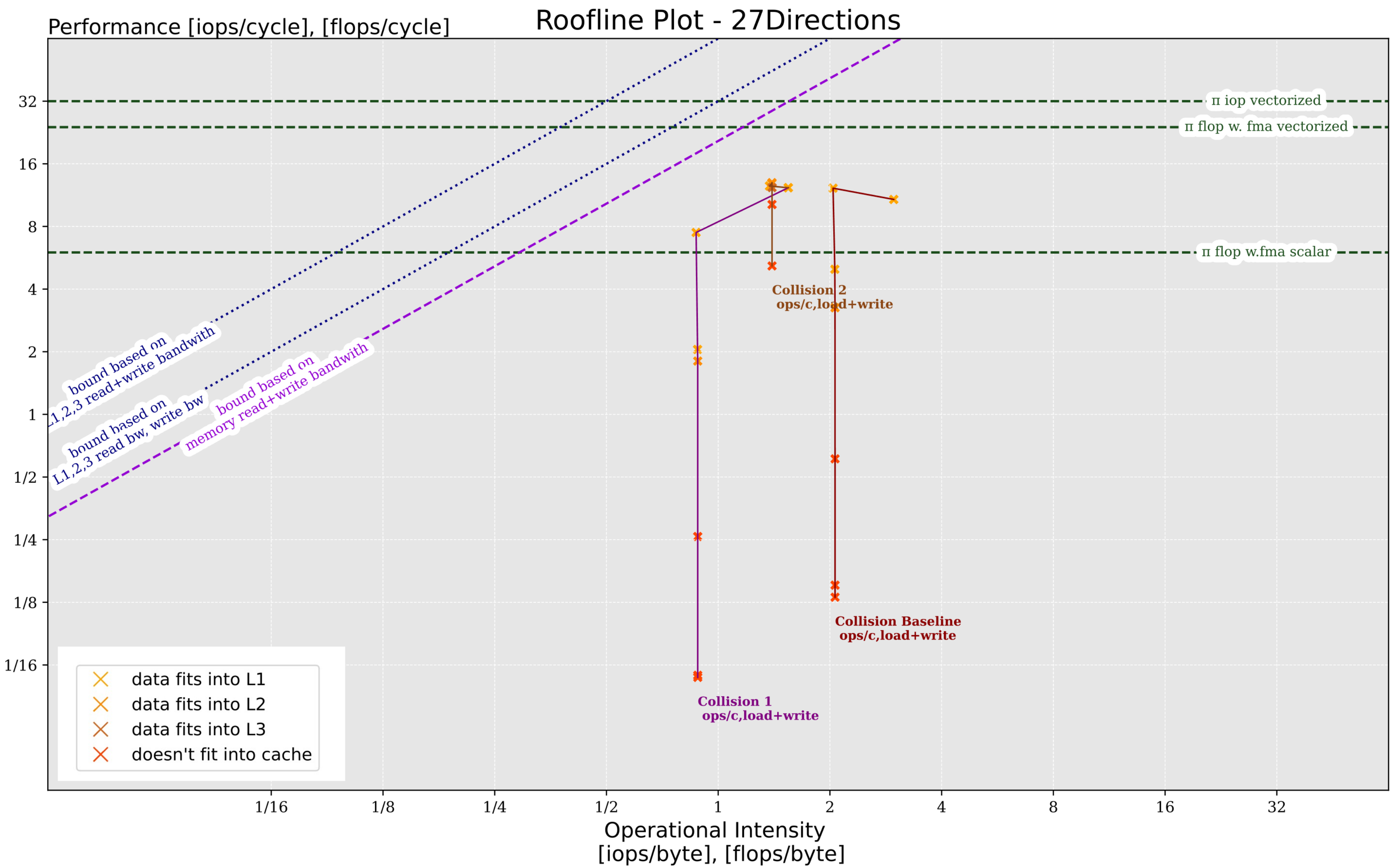
Optimization 2 – Couette BC



Optimization 2 – Lees Edwards BC



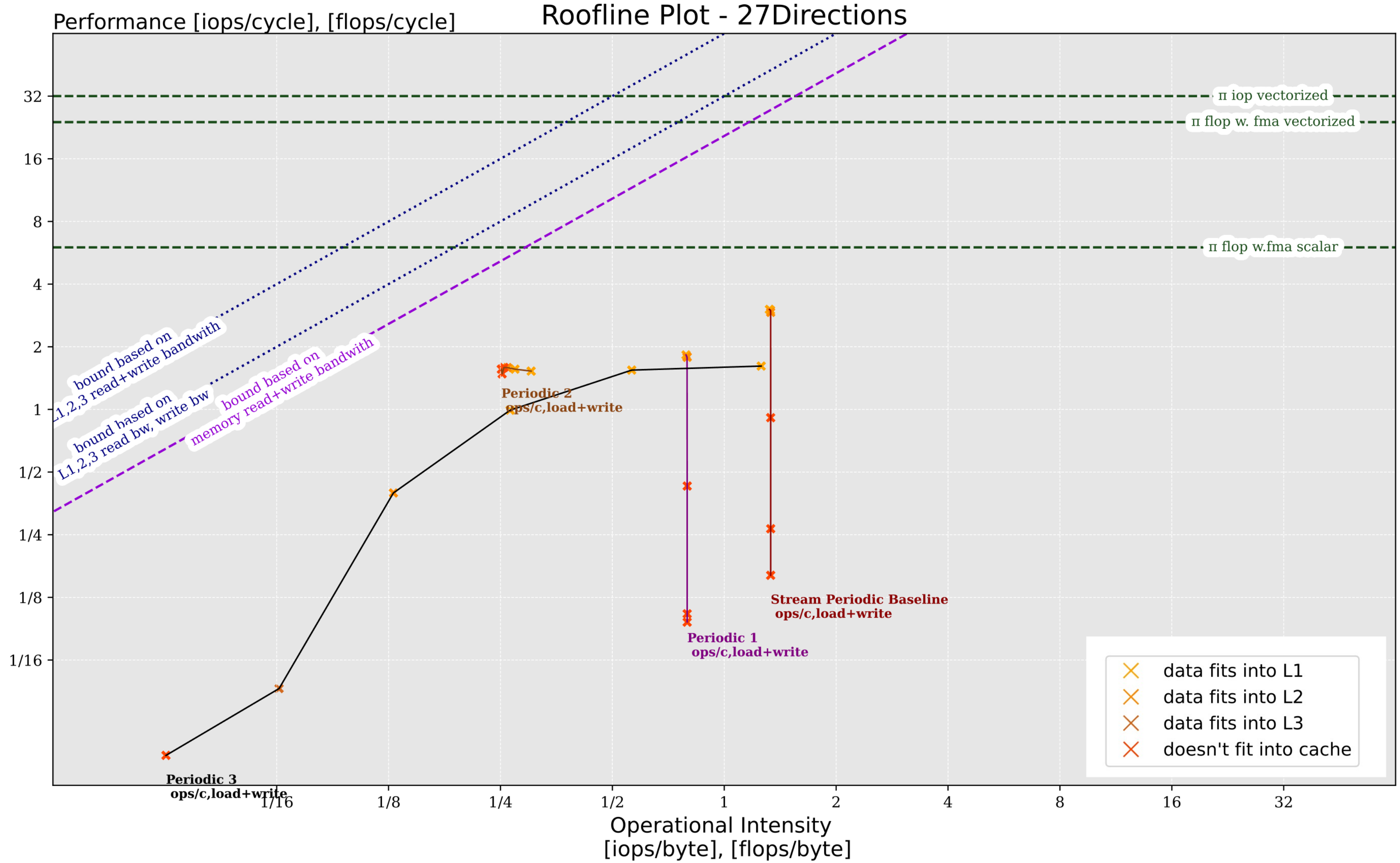
Optimization 2 - Collision



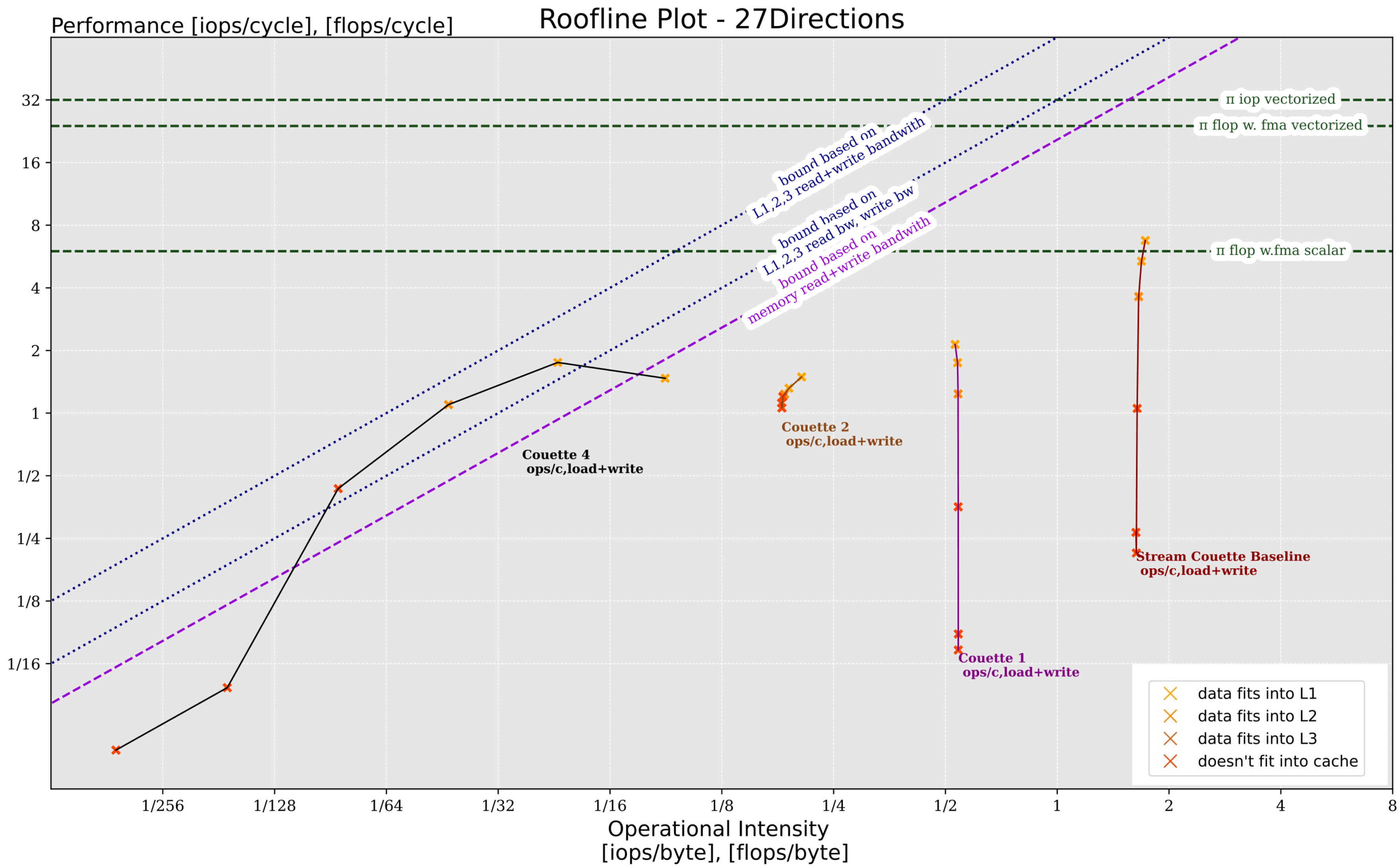
Optimization 3 + 4

SSA, memcpy, Memory Layout Adaptation, Vectorisation

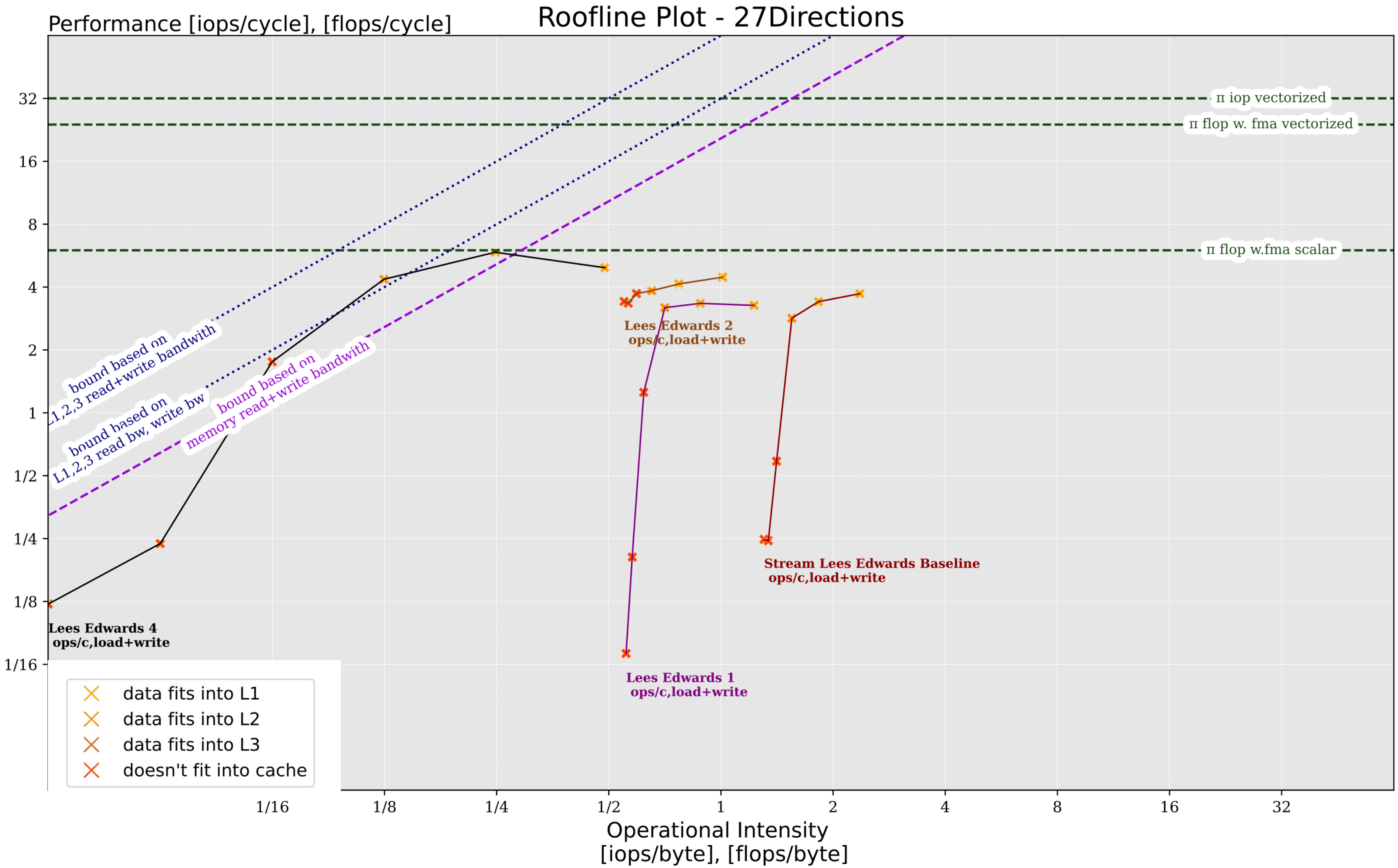
Optimization 3 & 4 – Periodic BC



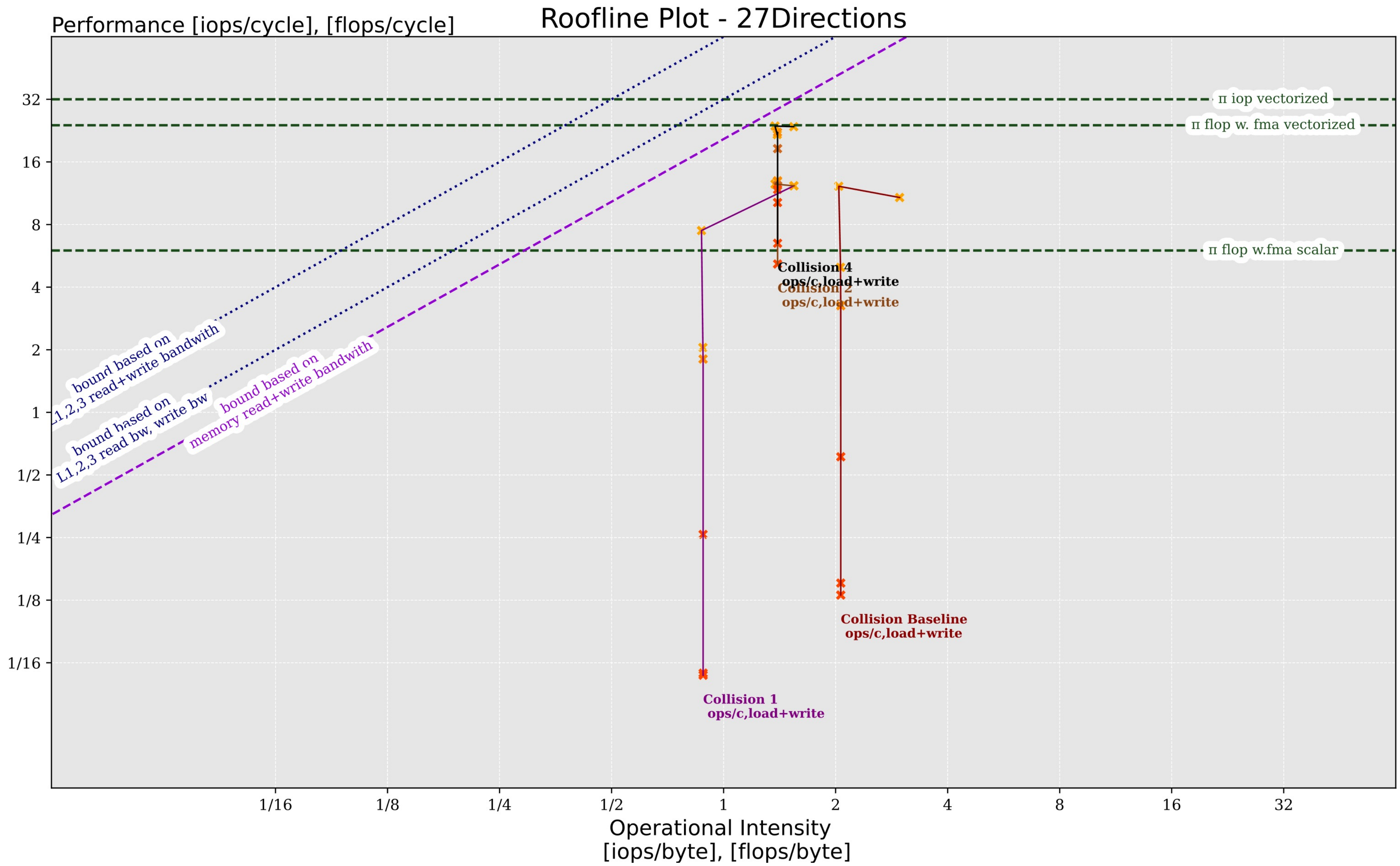
Optimization 3 & 4 – Couette BC



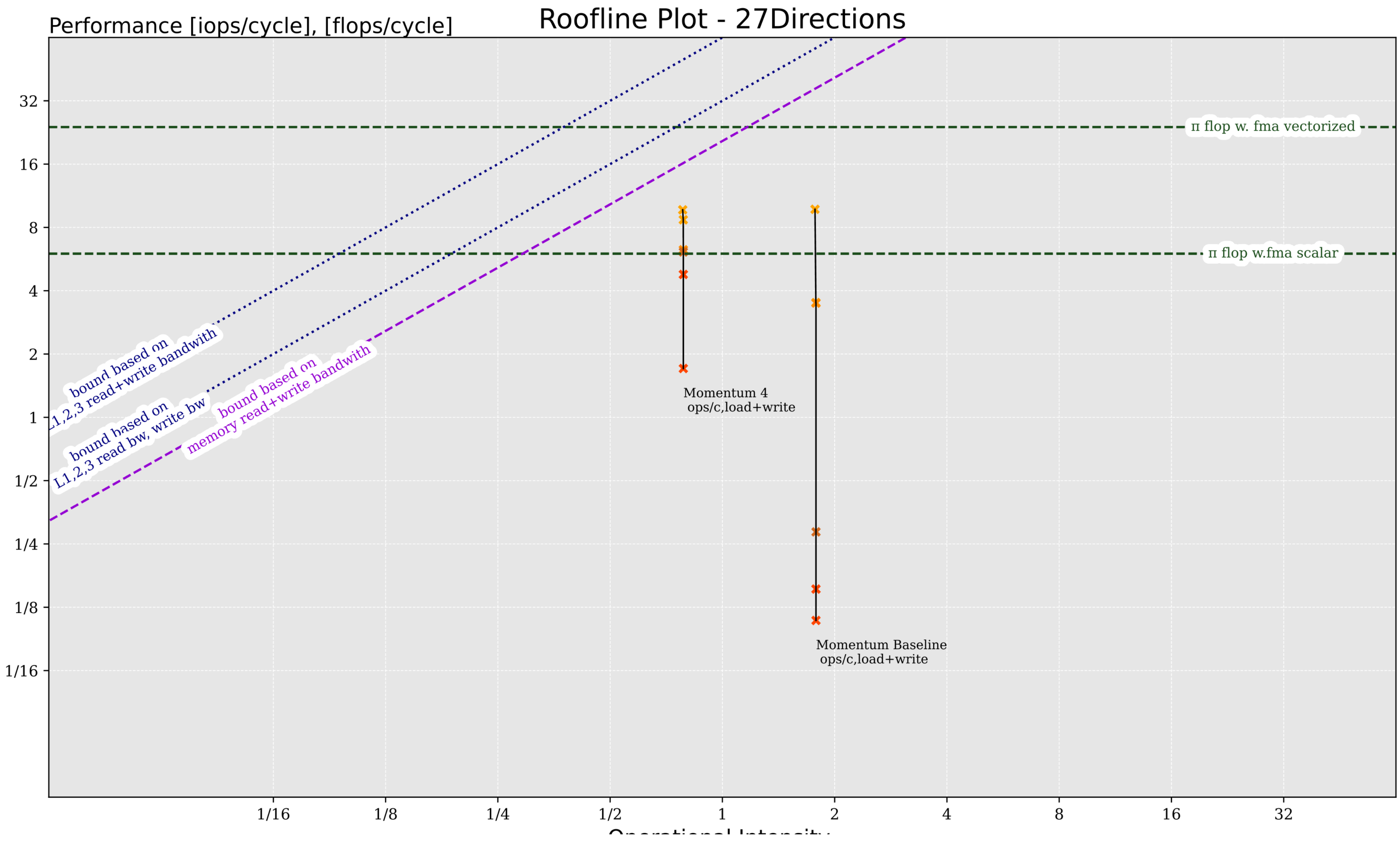
Optimization 3 & 4 – Lees Edwards BC



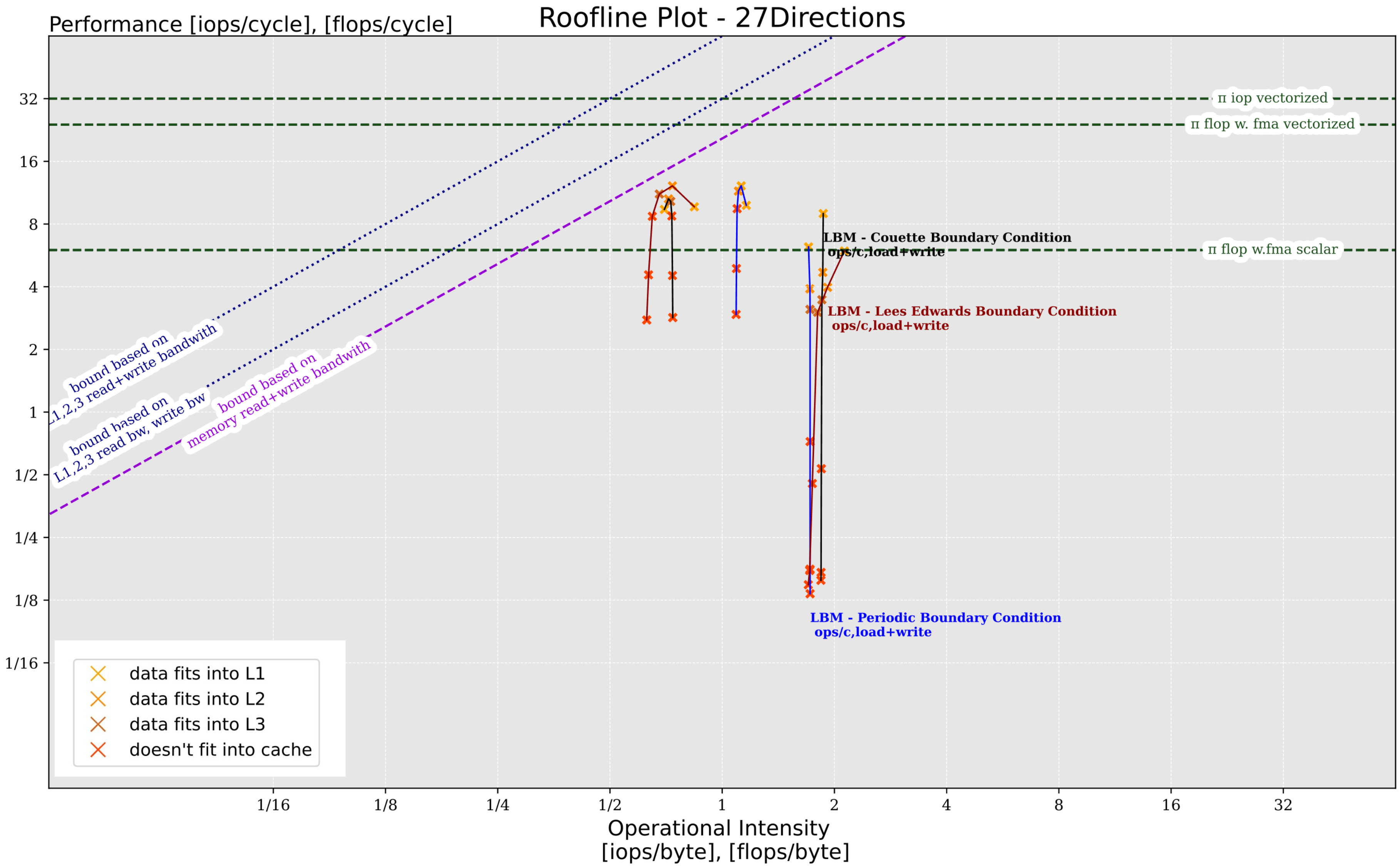
Optimization 3 & 4 - Collision



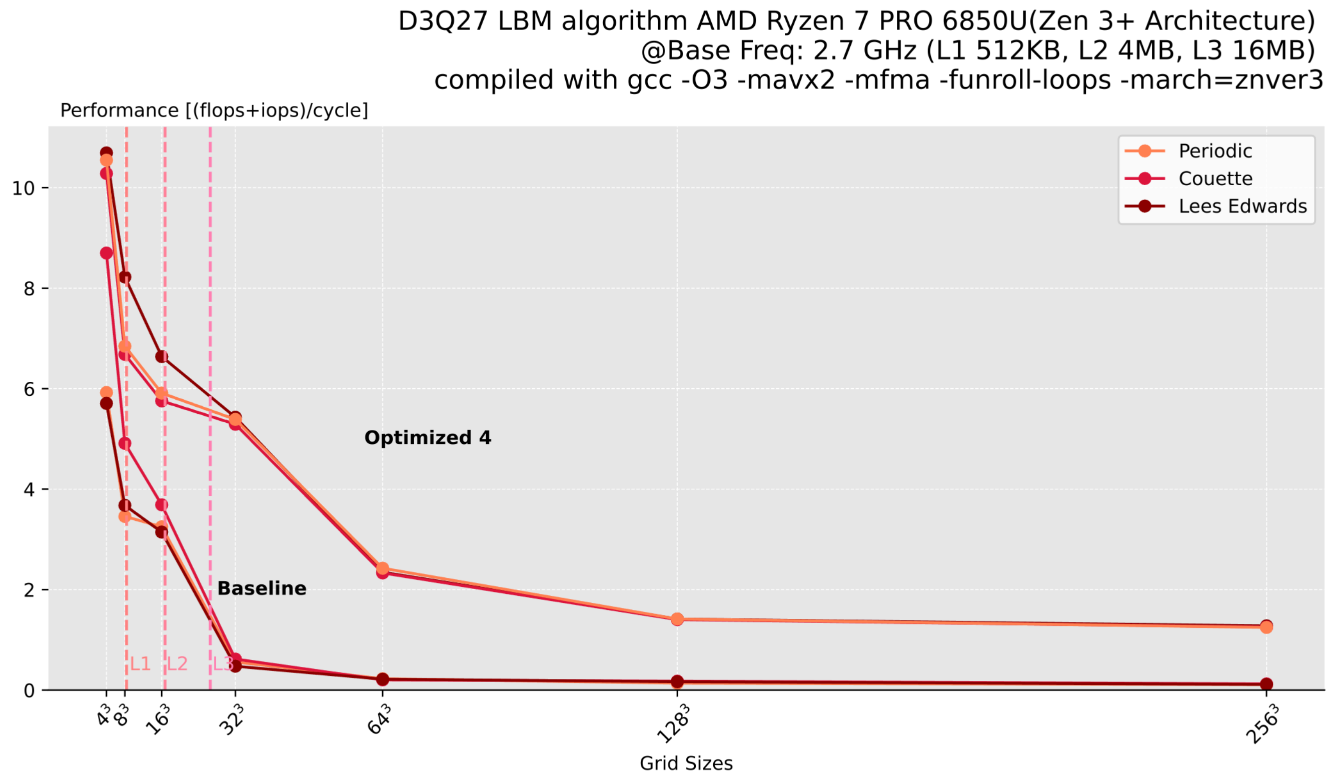
Optimisation 4 - Momentum



Roofline Overall Baseline and 4



Overall Results & Conclusion

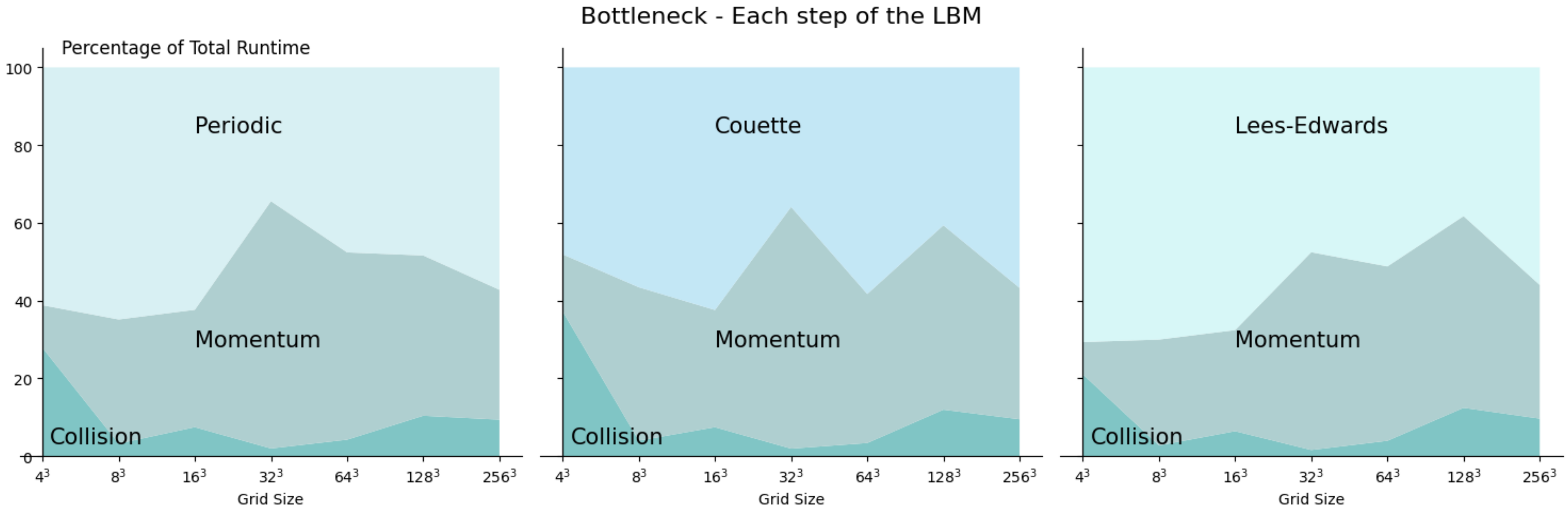


Speedup			
Grid	P	C	L
4	4.66	3.61	**4.81**
8	**9.62**	7.75	9.37
16	**11.6**	10.2	11.5
32	41.6	55.8	**60.5**
64	88.3	**90.9**	83.2
128	**69.8**	62.3	59.4

Extra Slides

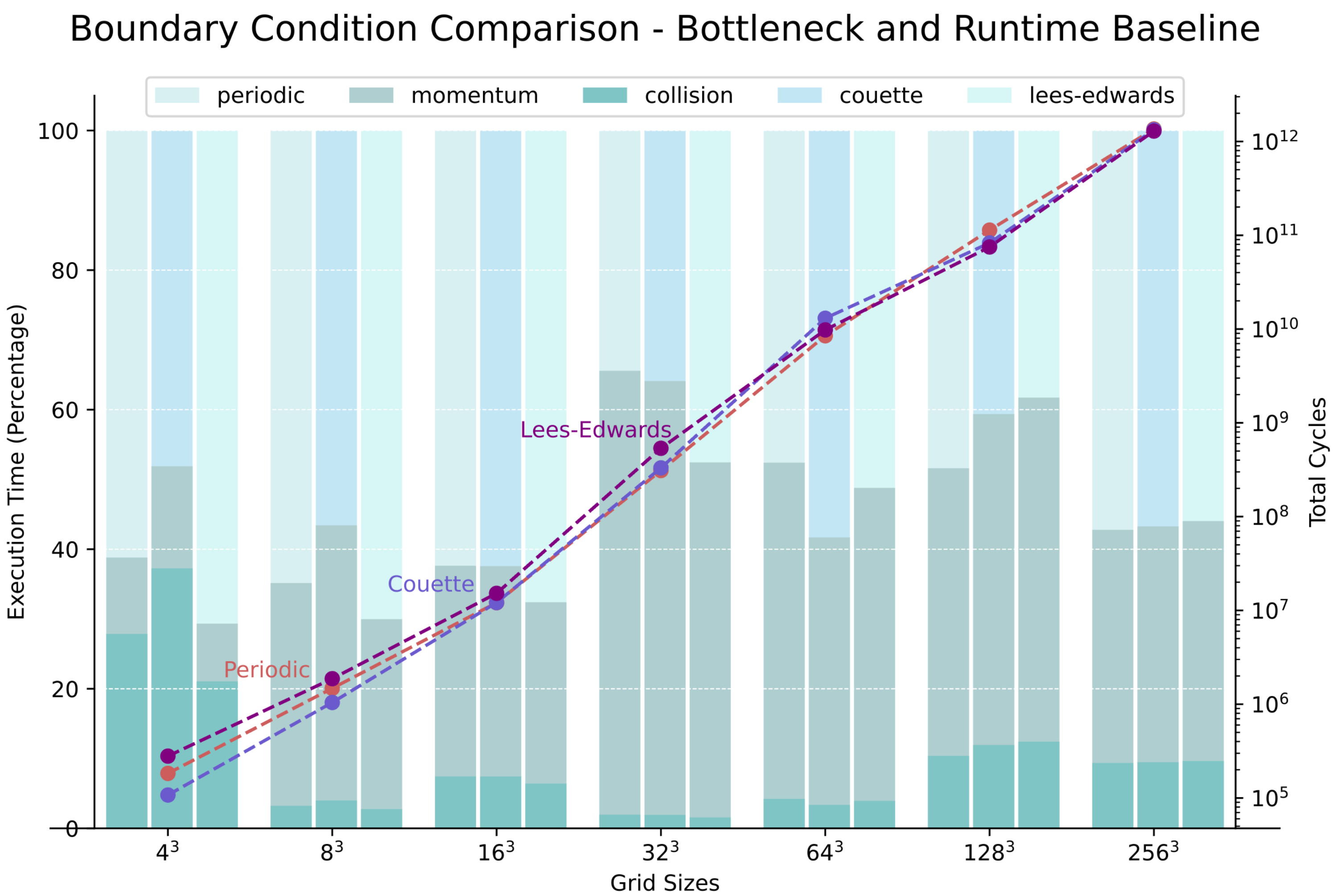
Boundary Condition Runtime vs Bottleneck Optimizations

Baseline



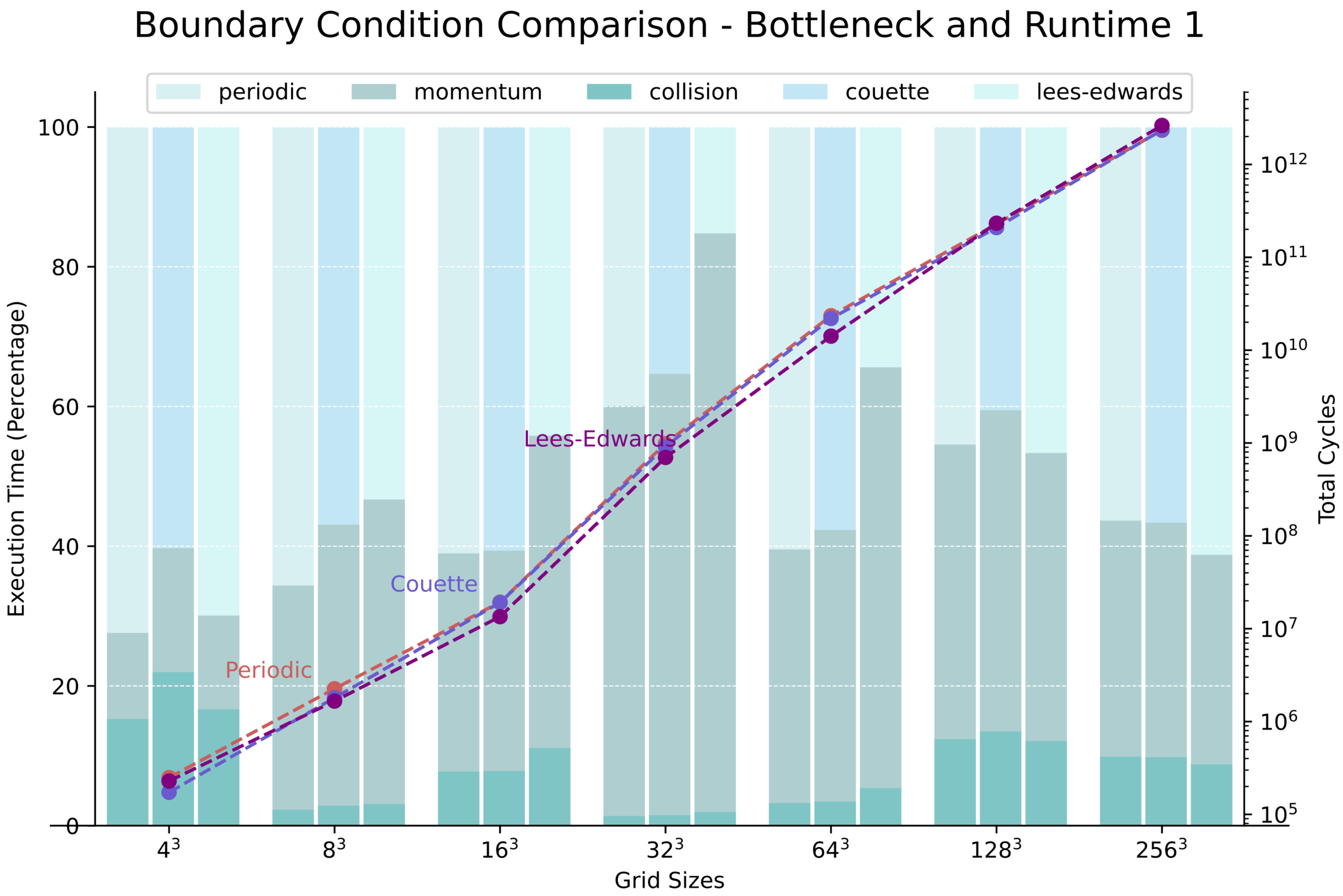
Bottleneck - Each step of the LBM
Percentage of Total Runtime
100
80
60
40
20
0
Periodic
Momentum
Collision
Couette
Momentum
Collision
Lees-Edwards
Momentum
Collision
4^3 8^3 16^3 32^3 64^3 128^3 256^3
Grid Size
Grid Size
Grid Size

Baseline

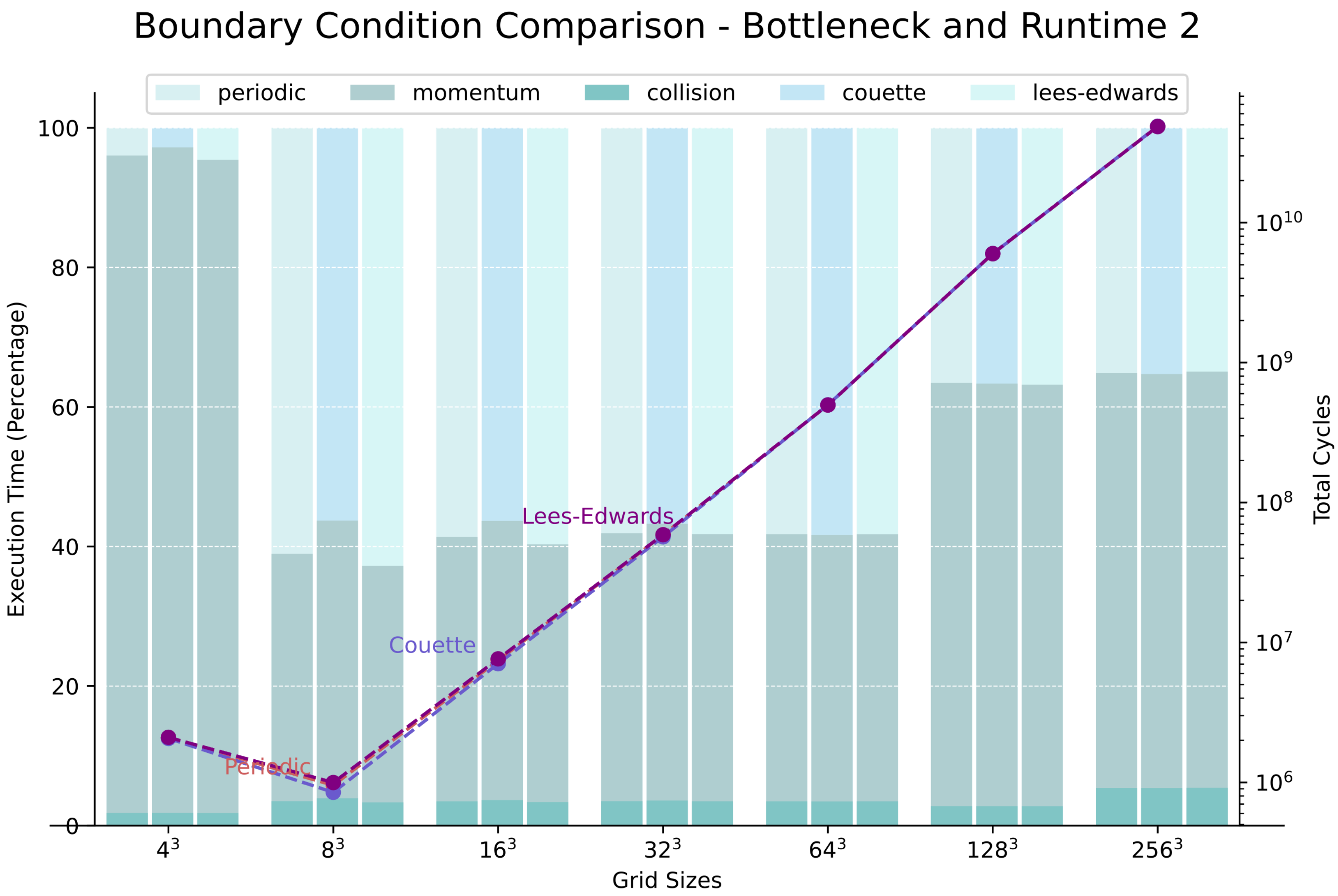


Boundary Condition Comparison - Bottleneck and Runtime Baseline
periodic
momentum
collision
couette
lees-edwards
Execution Time (Percentage)
Total Cycles
Grid Sizes
Lees-Edwards
Couette
Periodic
0
20
40
60
80
100
10^5
10^6
10^7
10^8
10^9
10^{10}
10^{11}
10^{12}
4^3
8^3
16^3
32^3
64^3
128^3
256^3

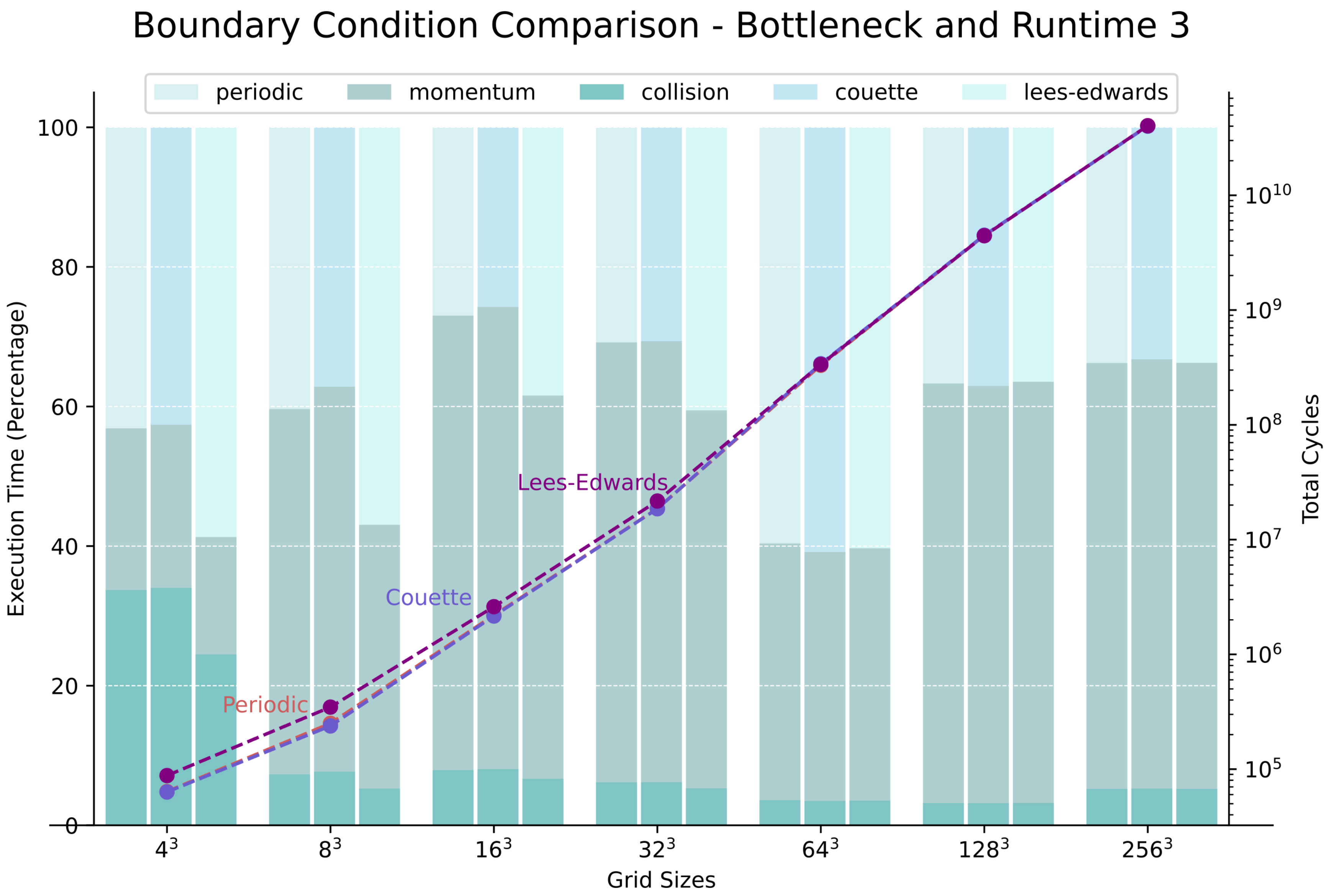
Optimization 1



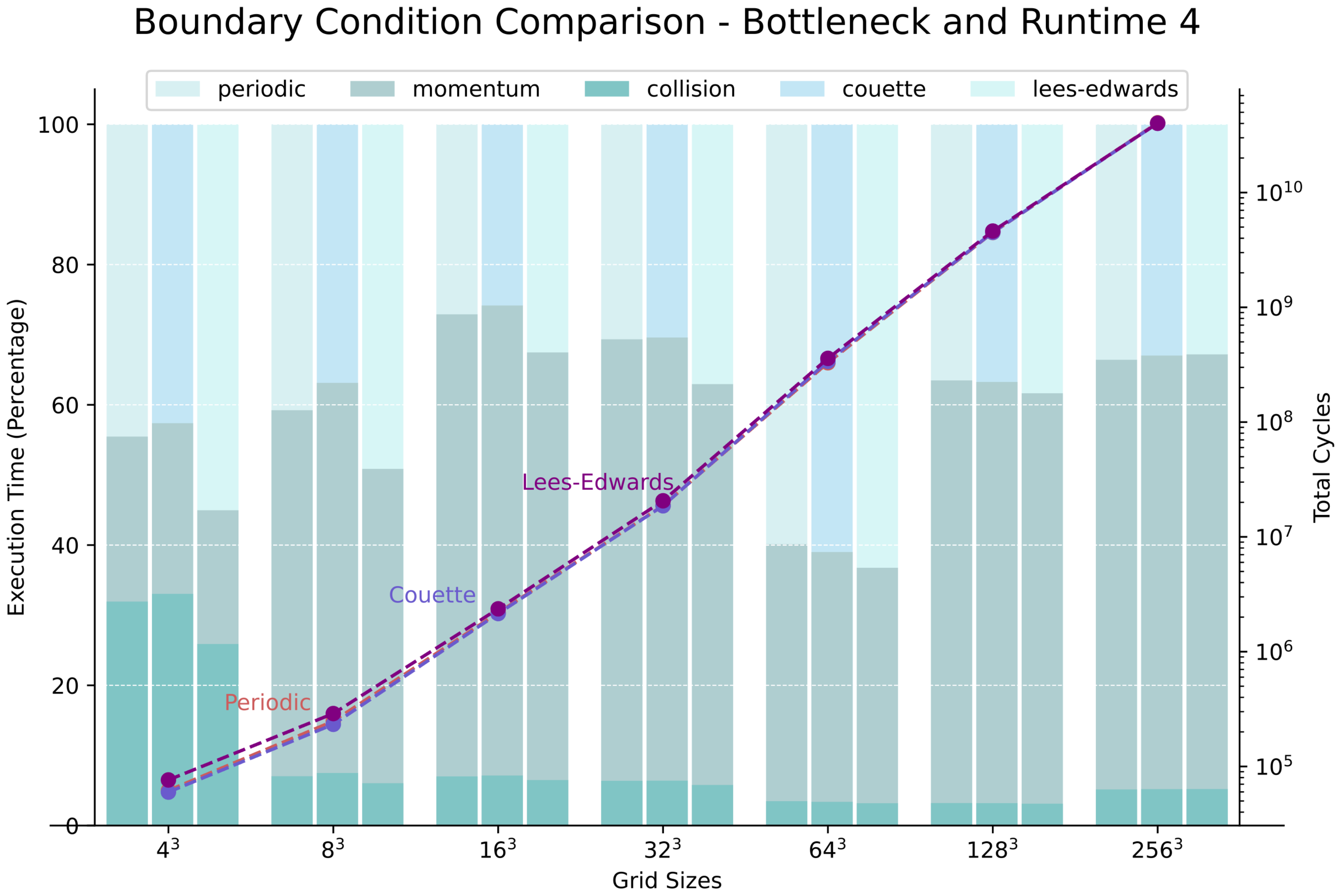
Optimization 2



Optimization 3

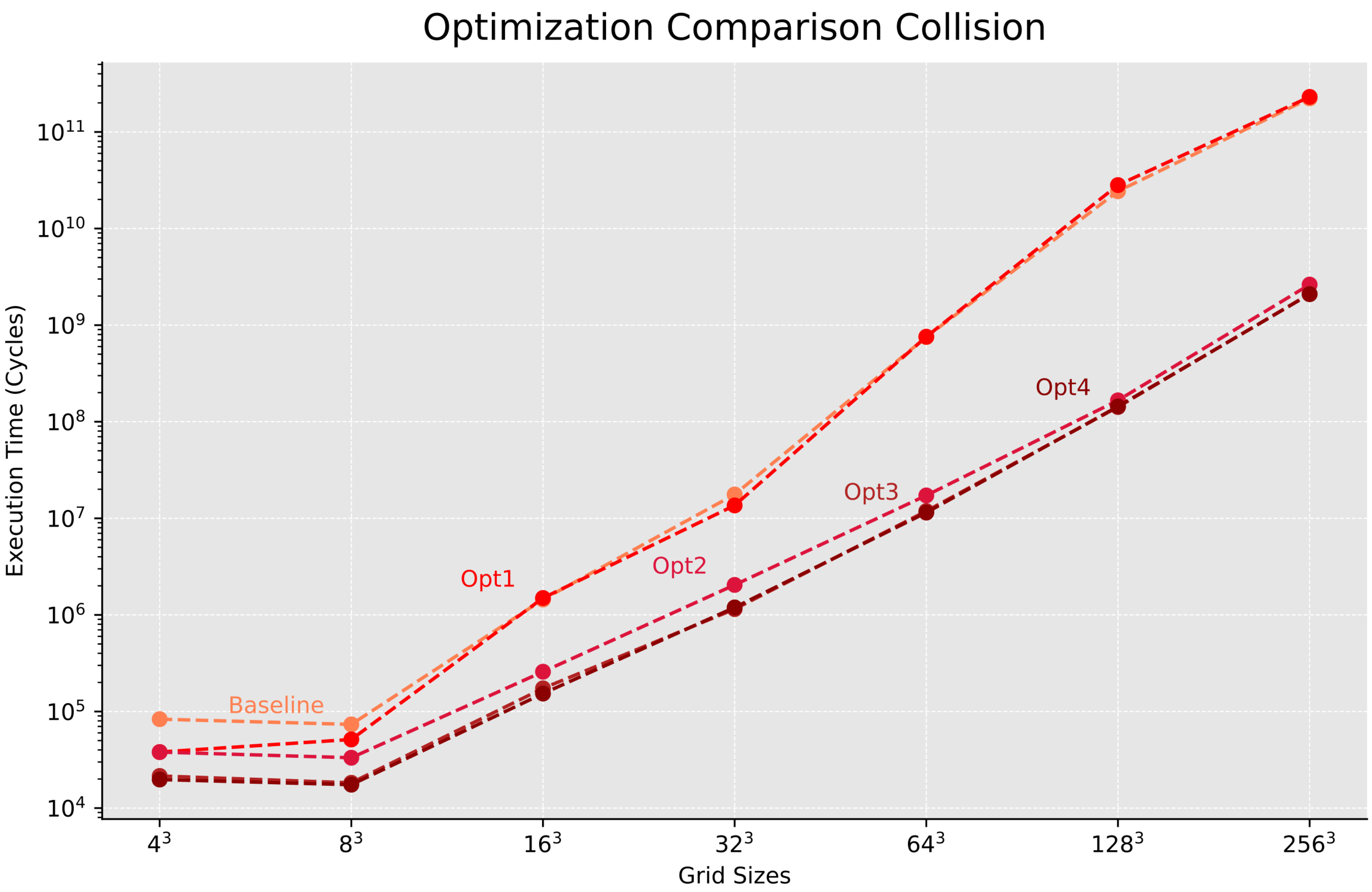


Optimization 4



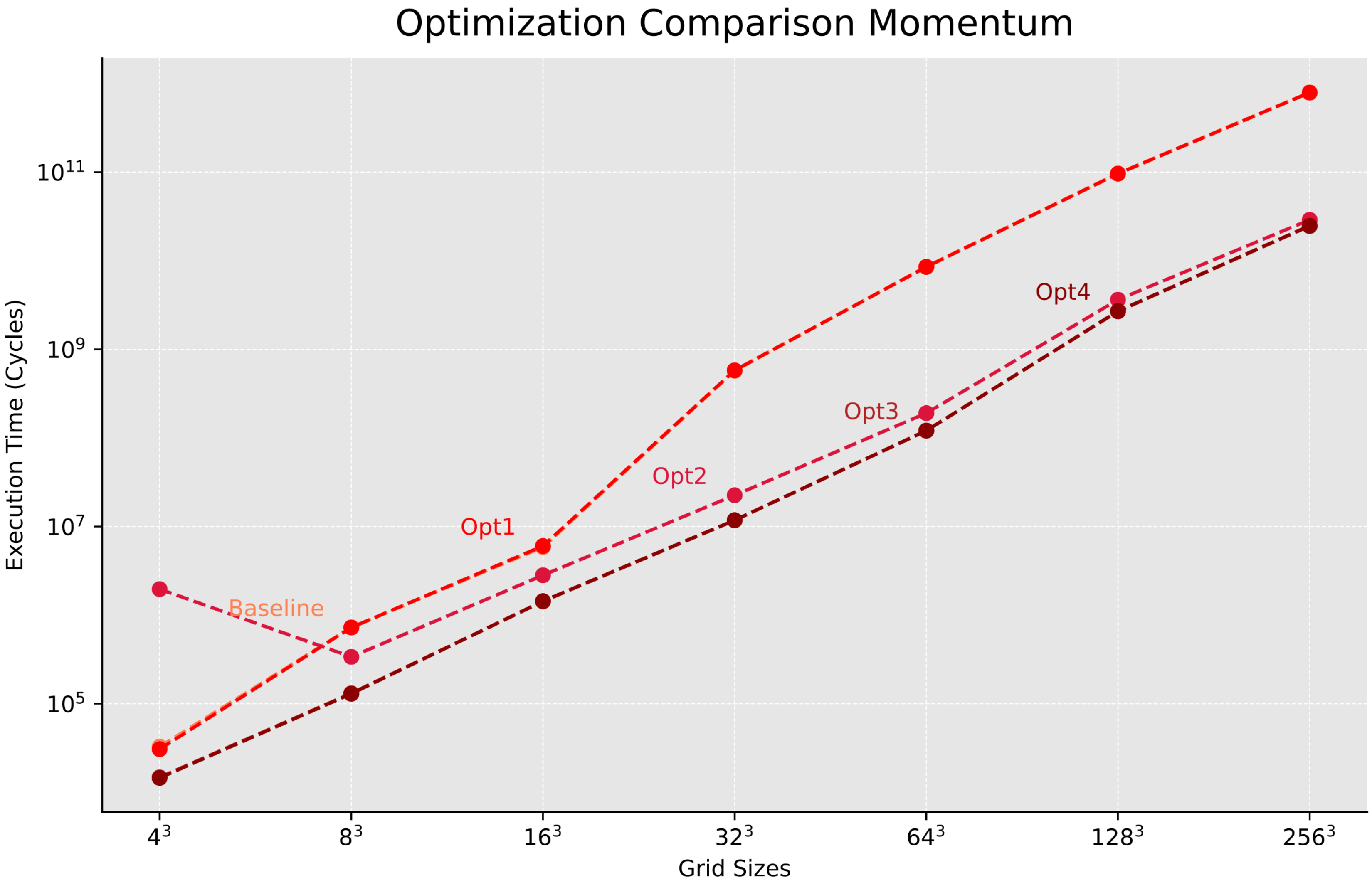
Optimization Comparison

Collision



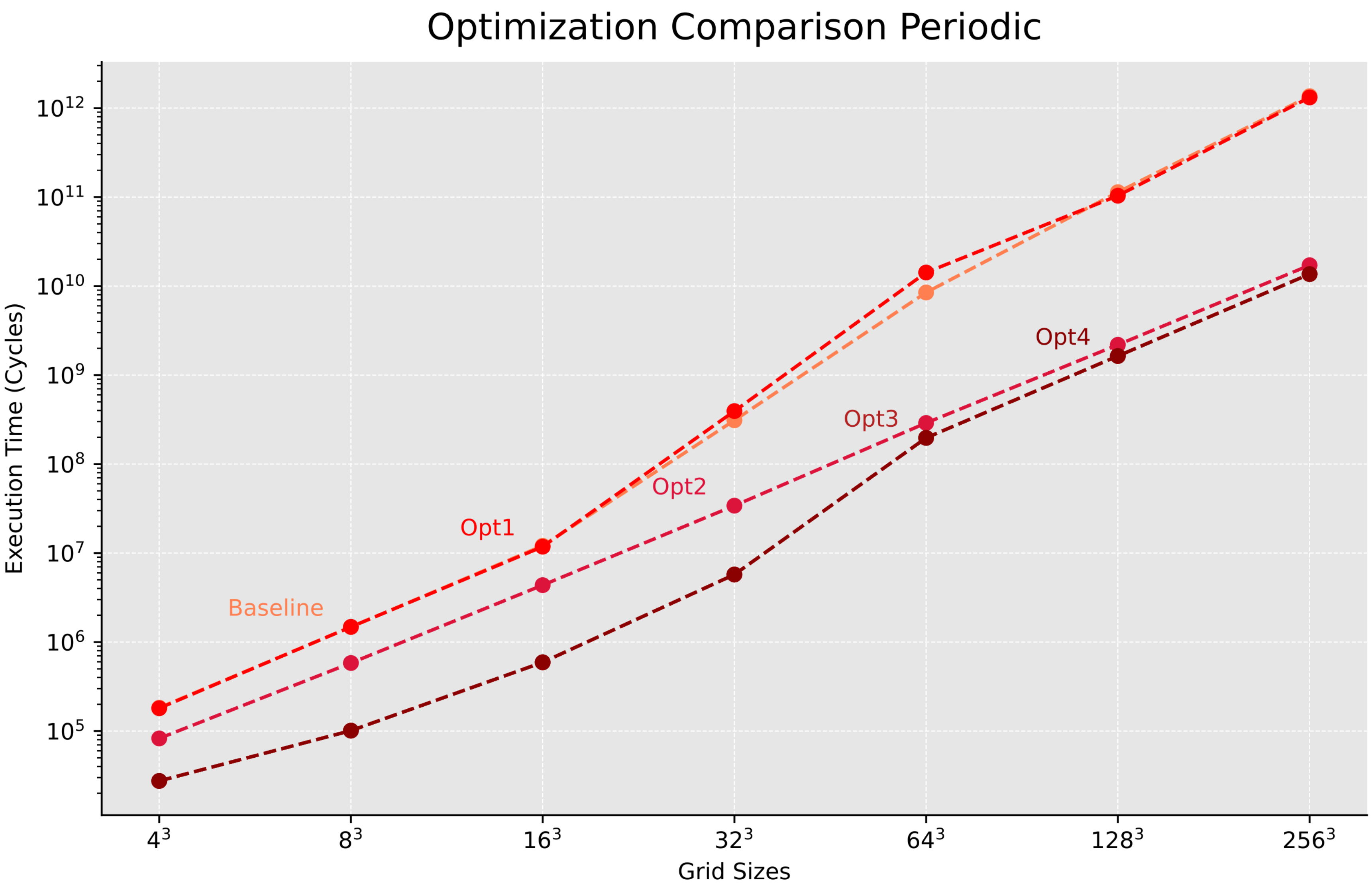
Optimization Comparison Collision
Execution Time (Cycles)
Grid Sizes
10^4
10^5
10^6
10^7
10^8
10^9
10^{10}
10^{11}
4^3
8^3
16^3
32^3
64^3
128^3
256^3
Baseline
Opt1
Opt2
Opt3
Opt4

Momentum



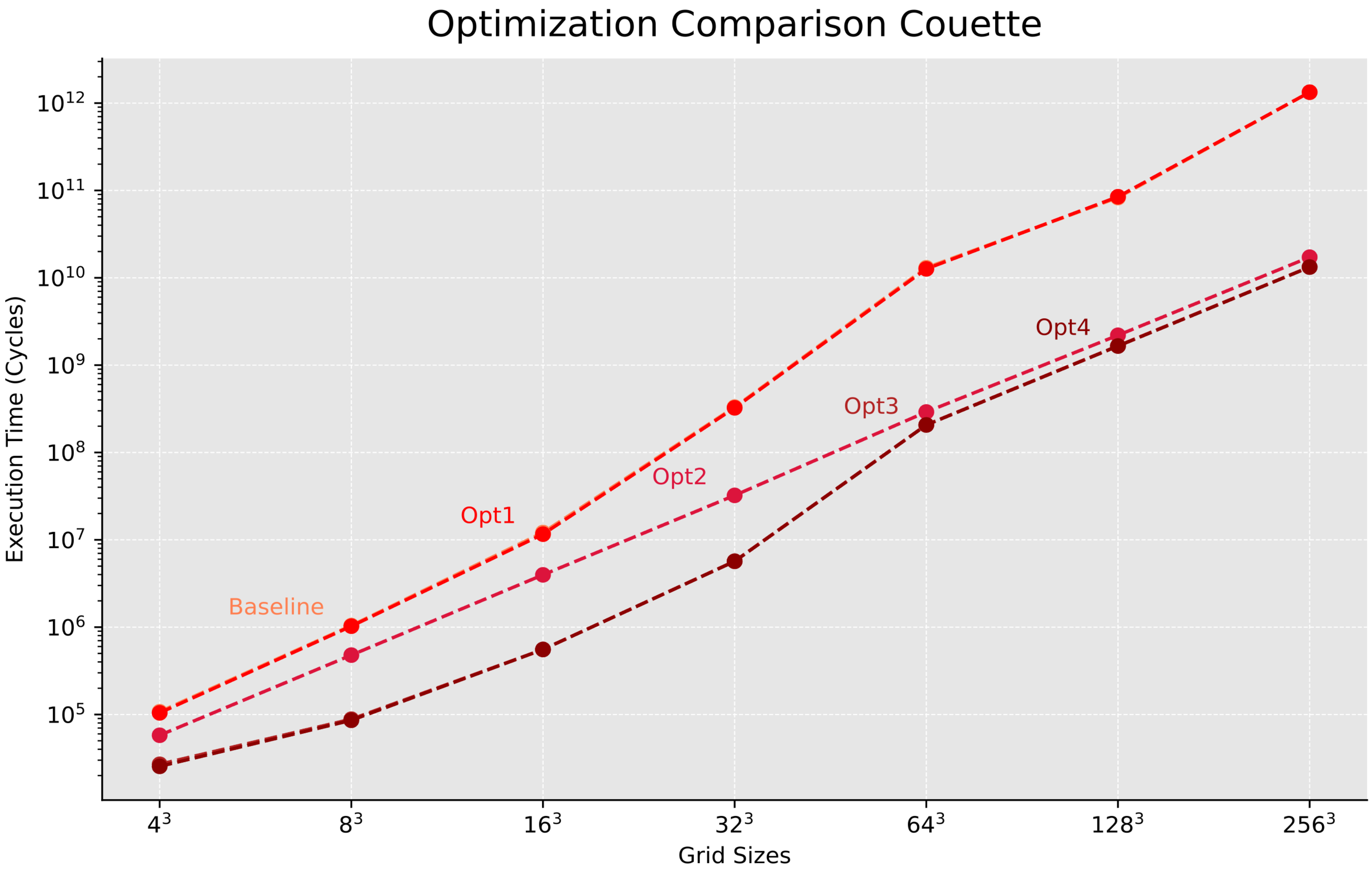
Optimization Comparison Momentum
Execution Time (Cycles)
10^{11}
10^{9}
10^{7}
10^{5}
Baseline
Opt1
Opt2
Opt3
Opt4
4^3
8^3
16^3
32^3
64^3
128^3
256^3
Grid Sizes

Periodic



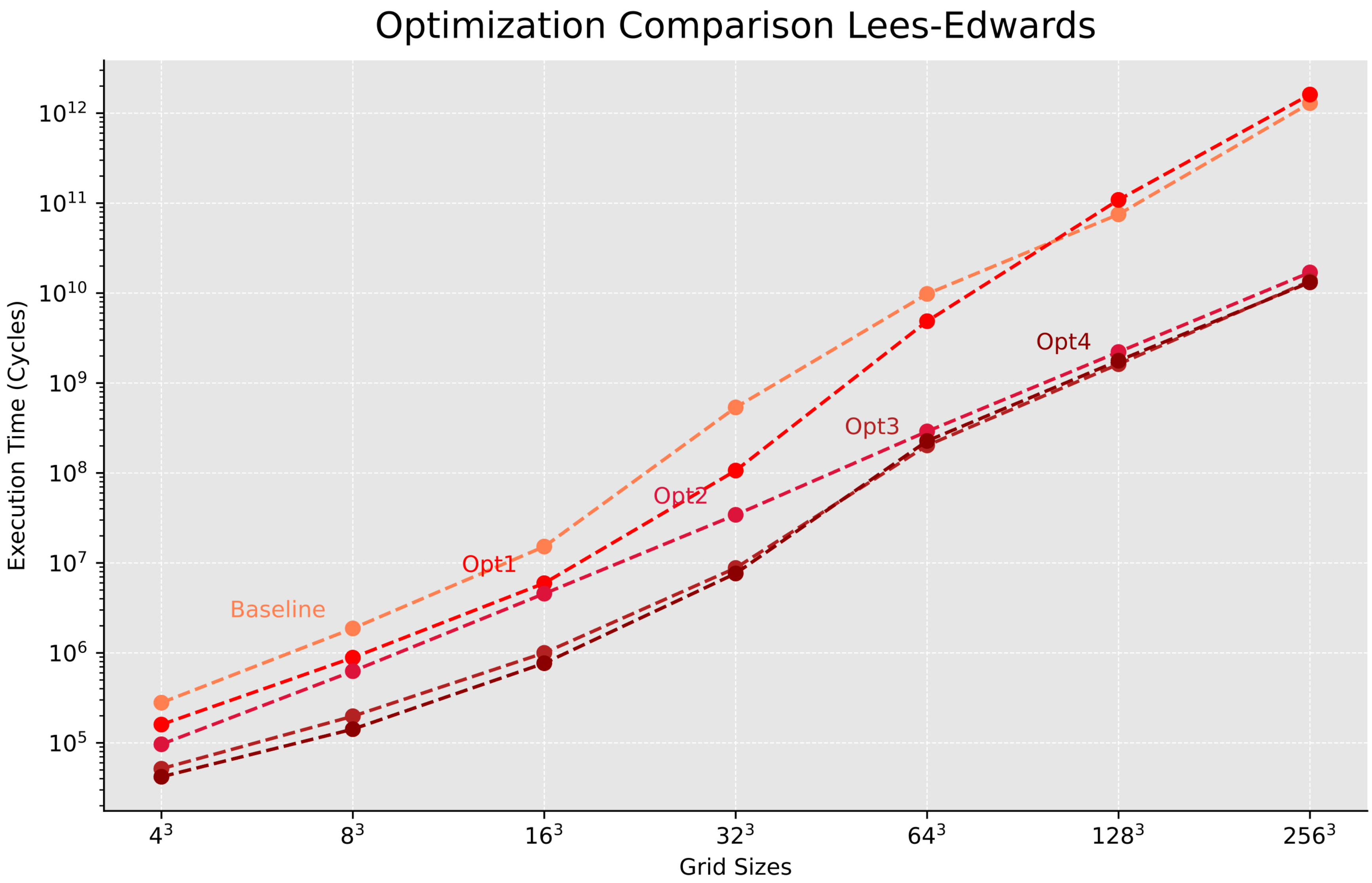
Optimization Comparison Periodic
Execution Time (Cycles)
10^12
10^11
10^10
10^9
10^8
10^7
10^6
10^5
Baseline
Opt1
Opt2
Opt3
Opt4
4^3
8^3
16^3
32^3
64^3
128^3
256^3
Grid Sizes

Couette

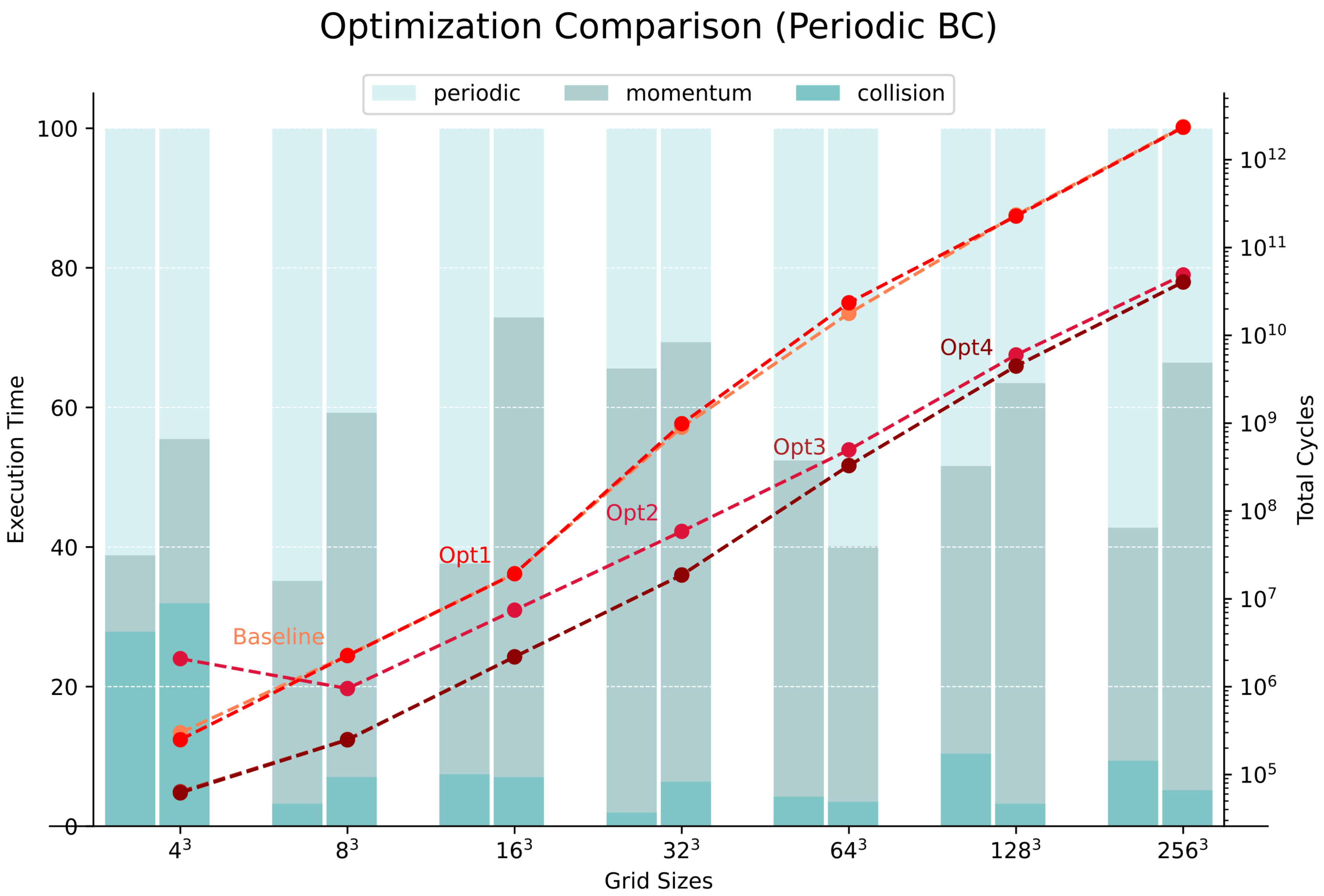


Optimization Comparison Couette
Execution Time (Cycles)
10^{12}
10^{11}
10^{10}
10^{9}
10^{8}
10^{7}
10^{6}
10^{5}
Baseline
Opt1
Opt2
Opt3
Opt4
4^3
8^3
16^3
32^3
64^3
128^3
256^3
Grid Sizes

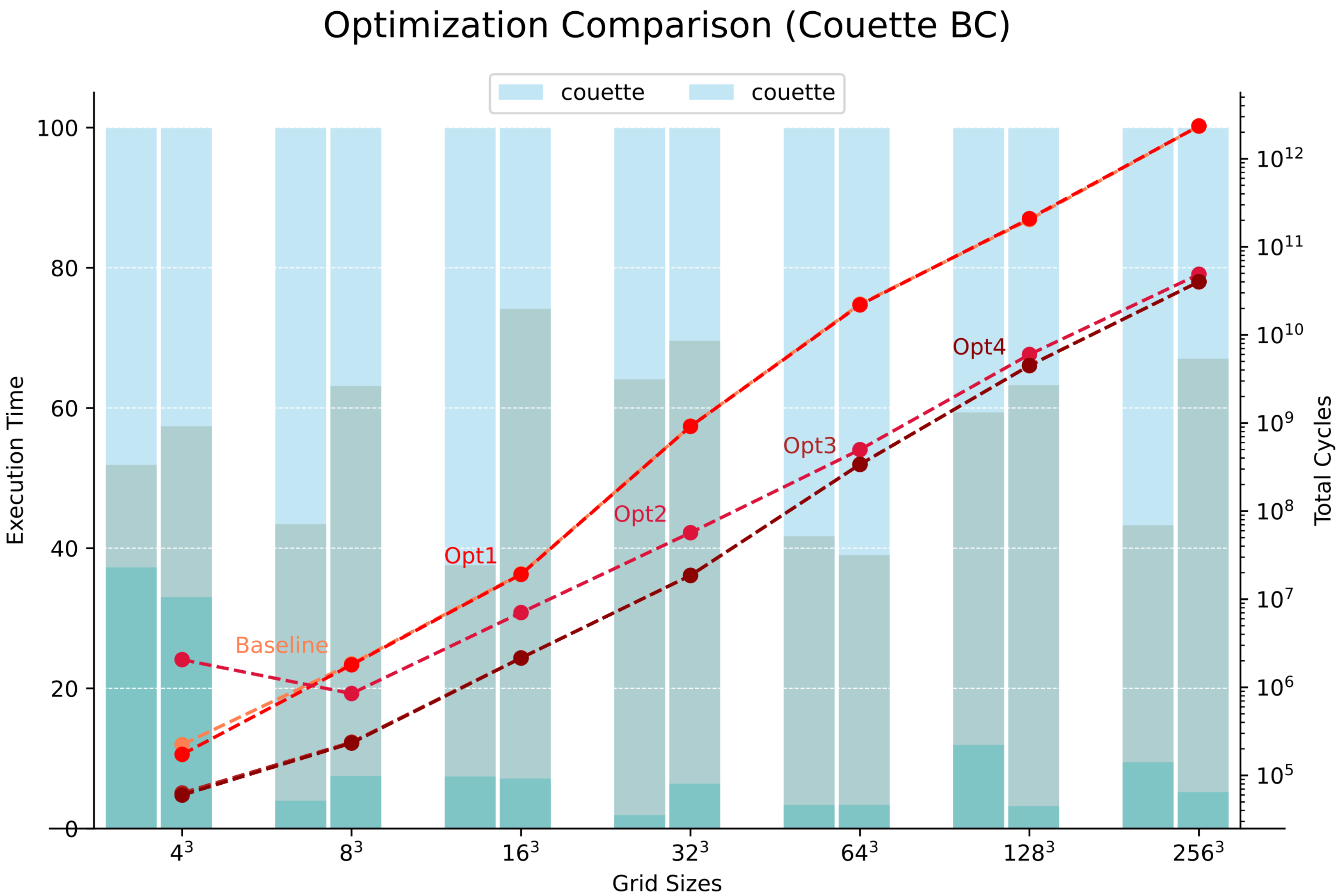
Lees Edwards



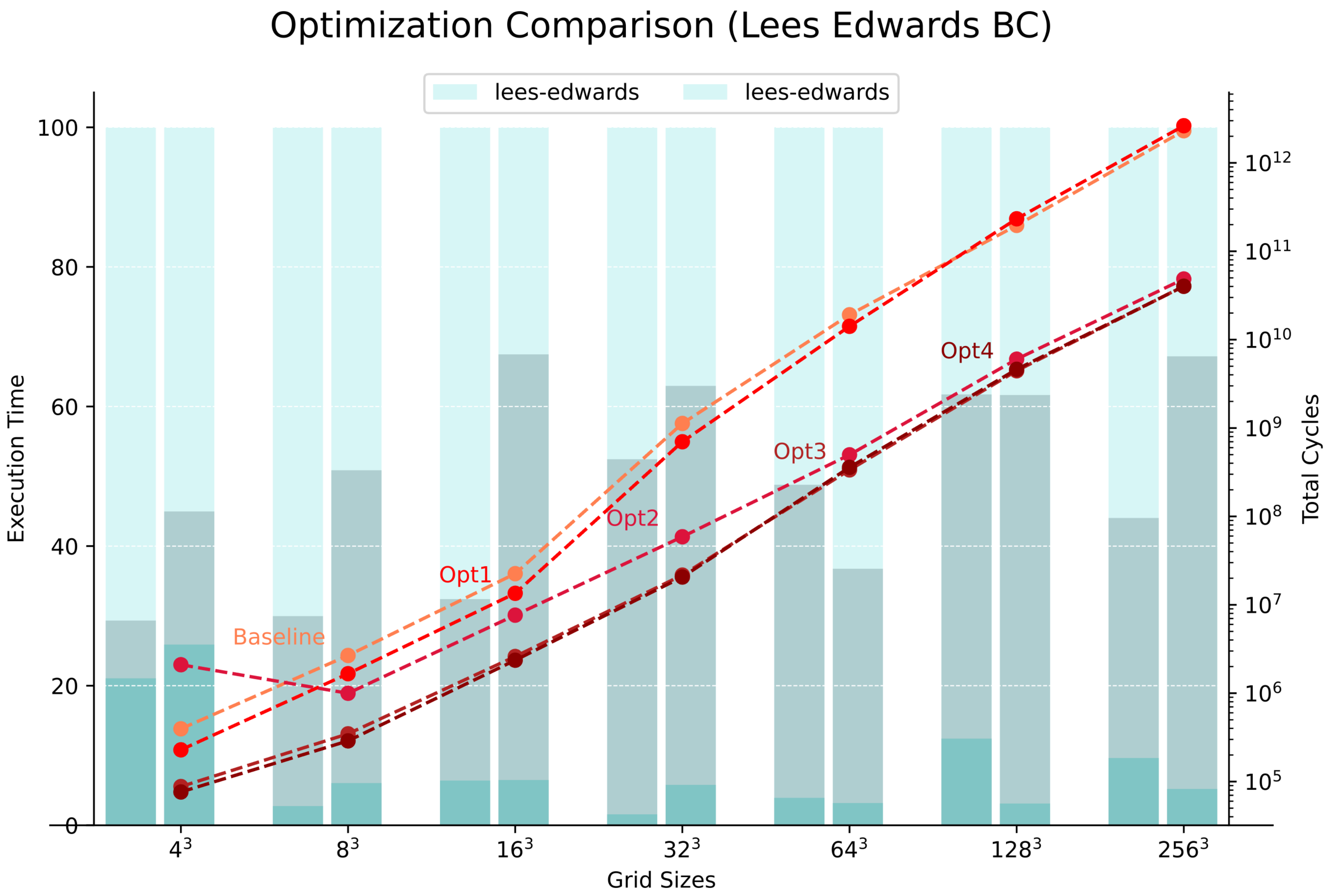
Total (Periodic BC)



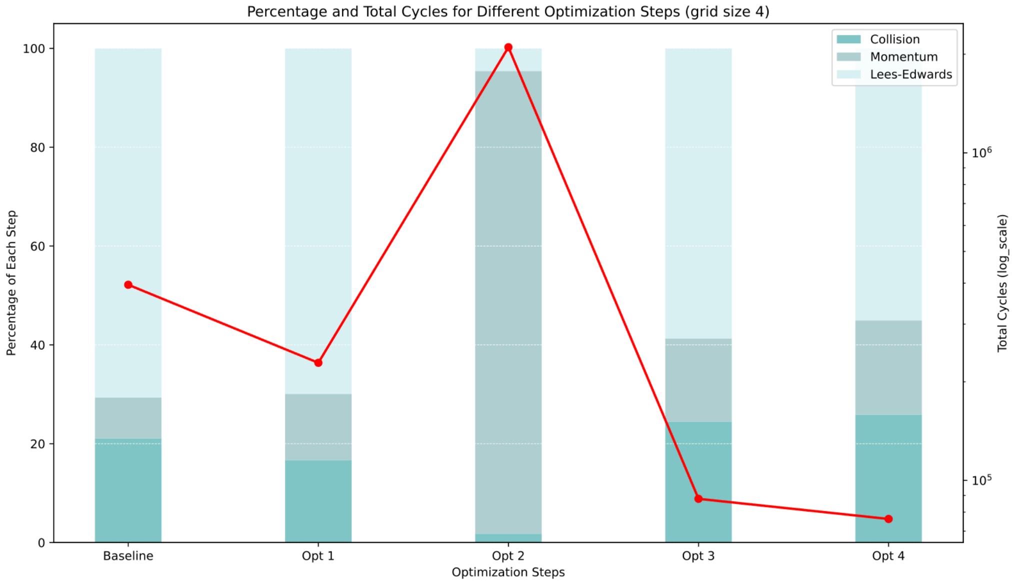
Total (Couette)

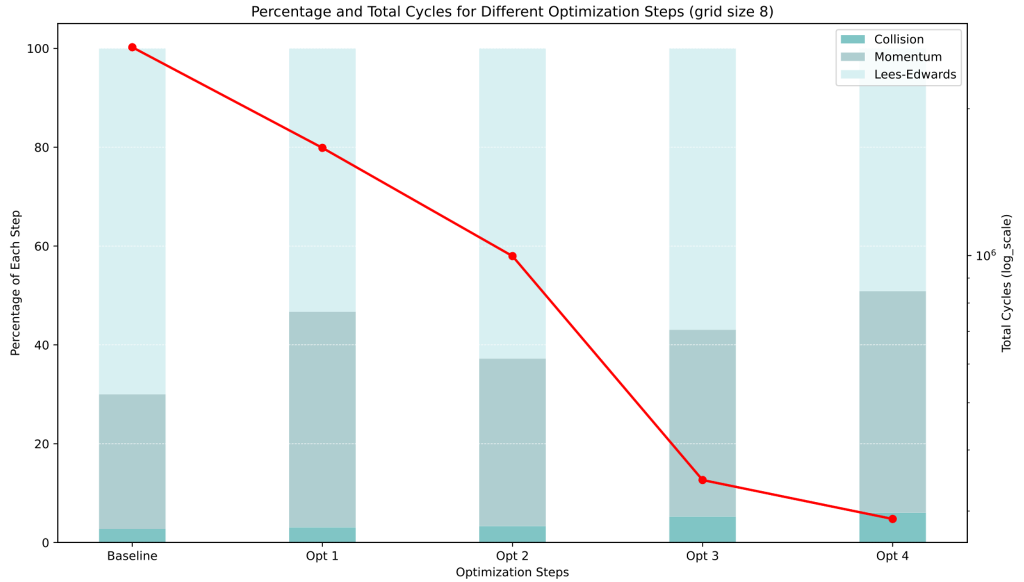


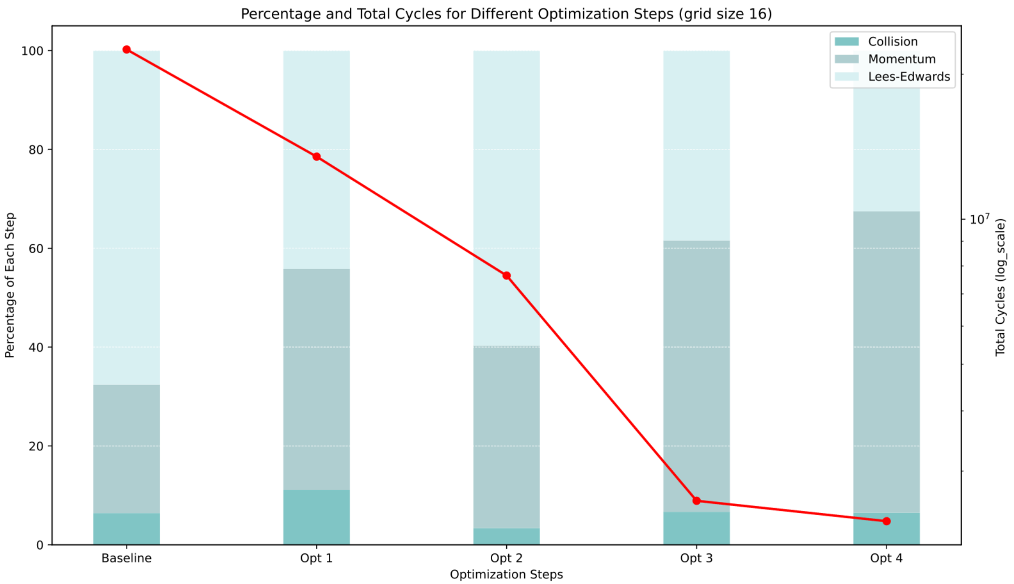
Total (Lees Edwards)

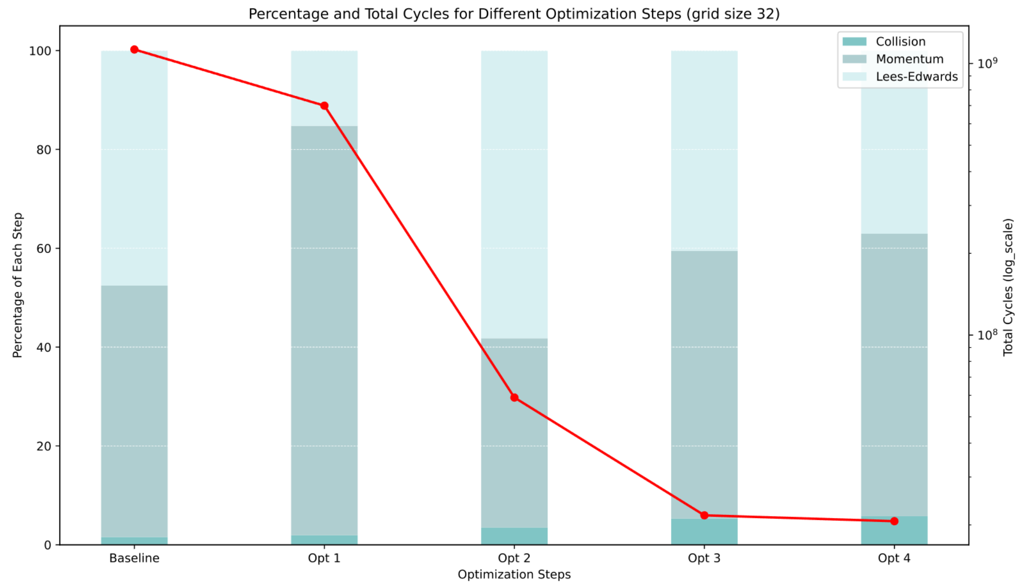


Different grid sizes

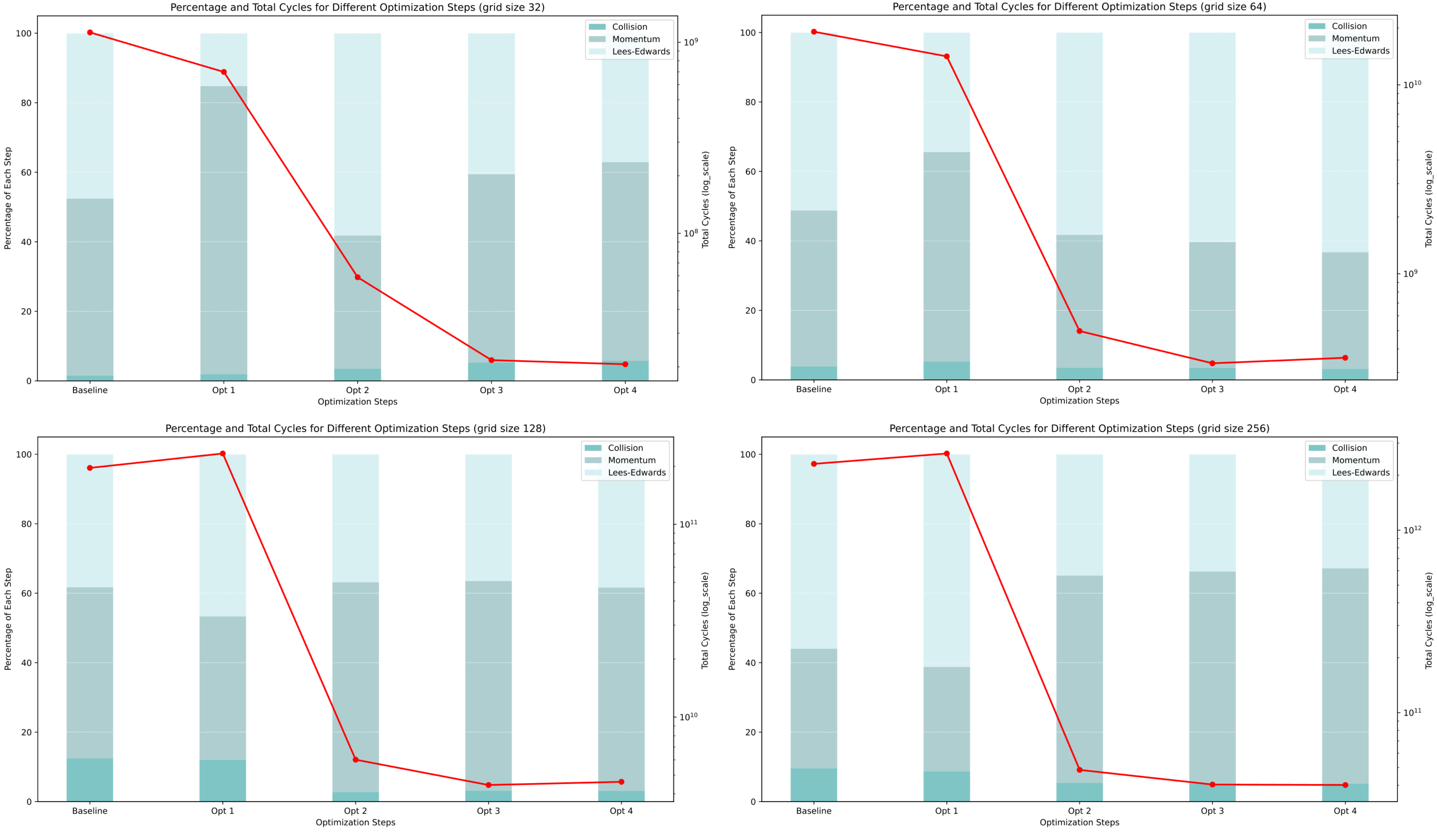






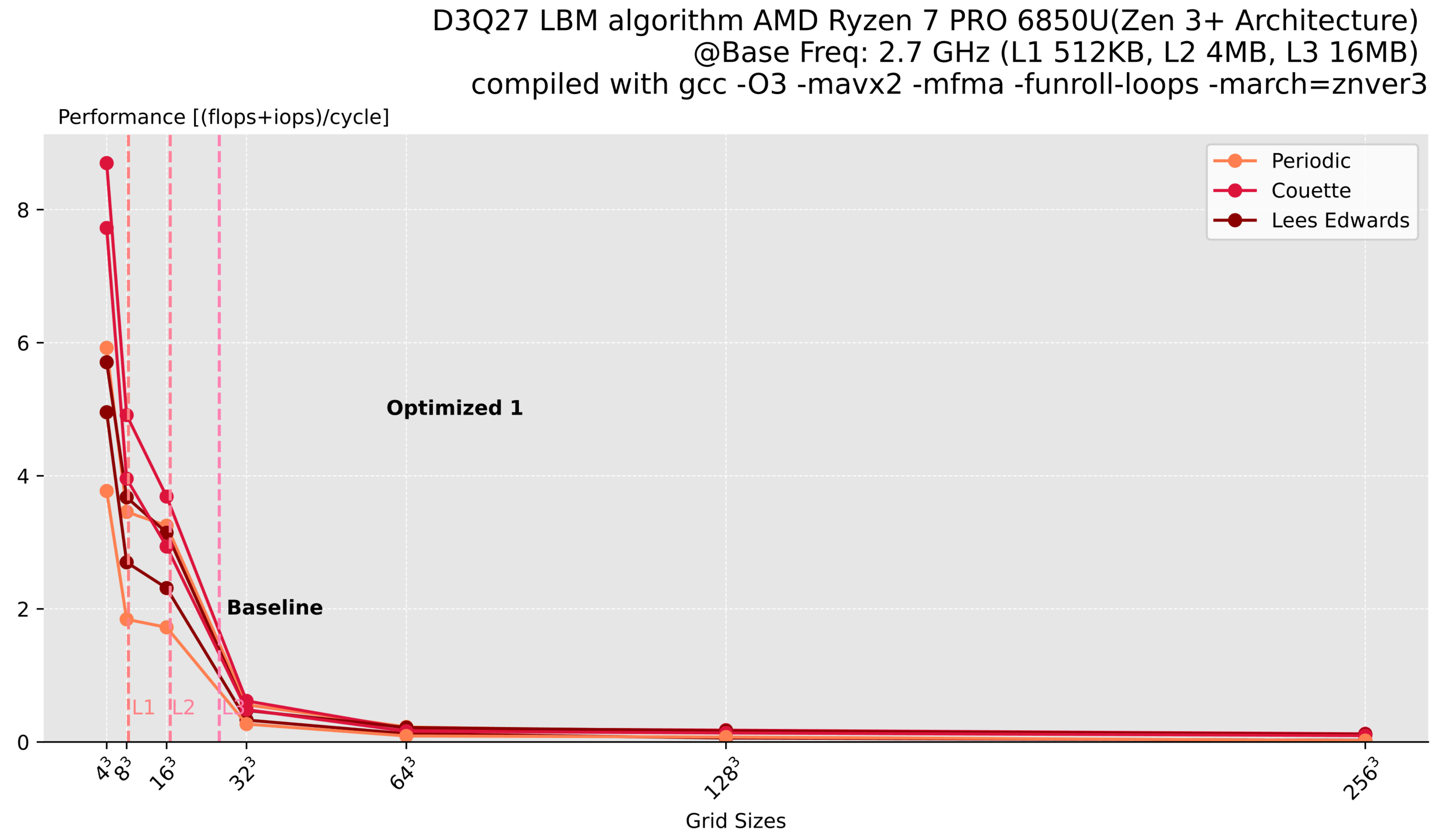


Different grid sizes

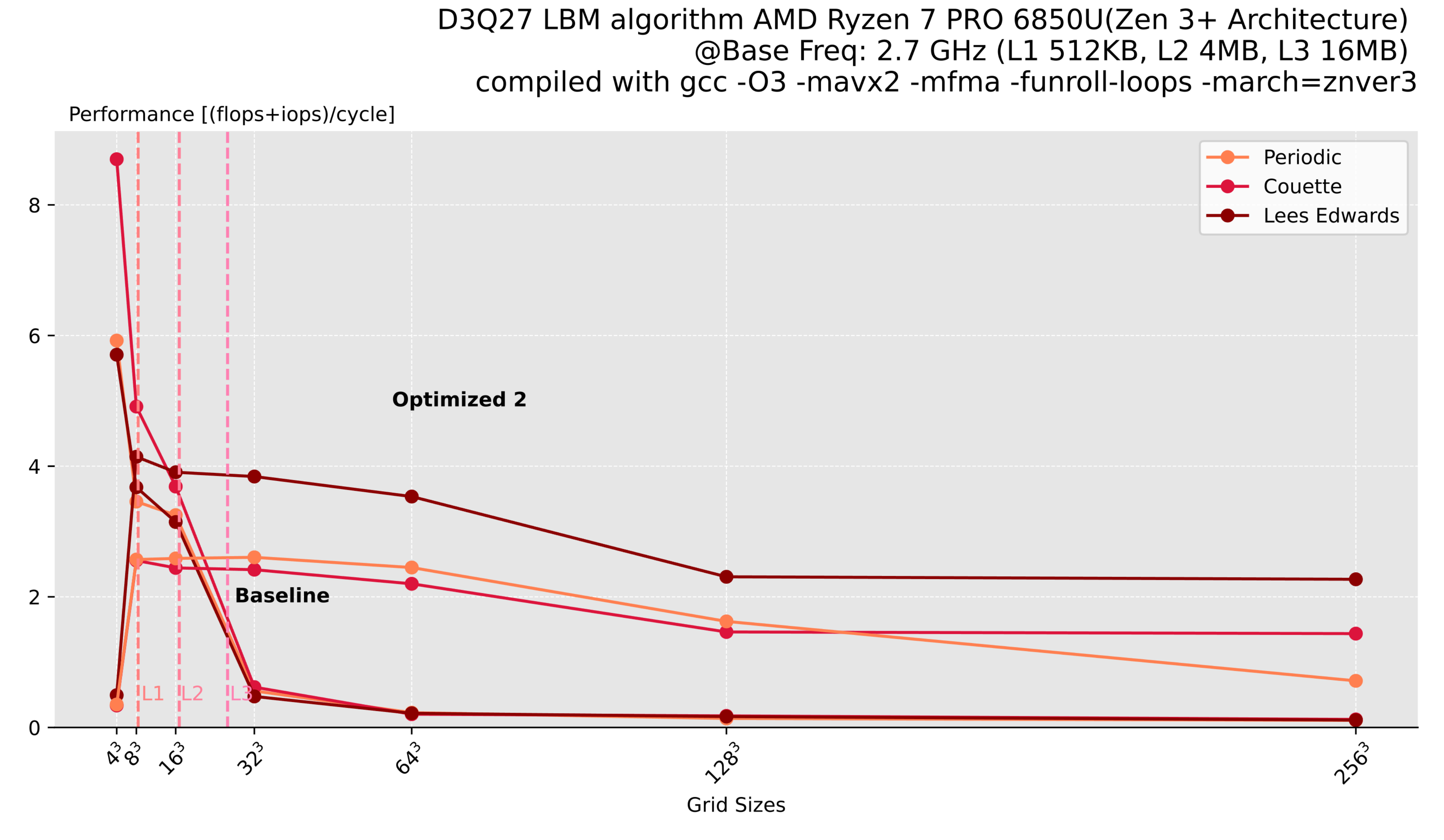


Performance Plots

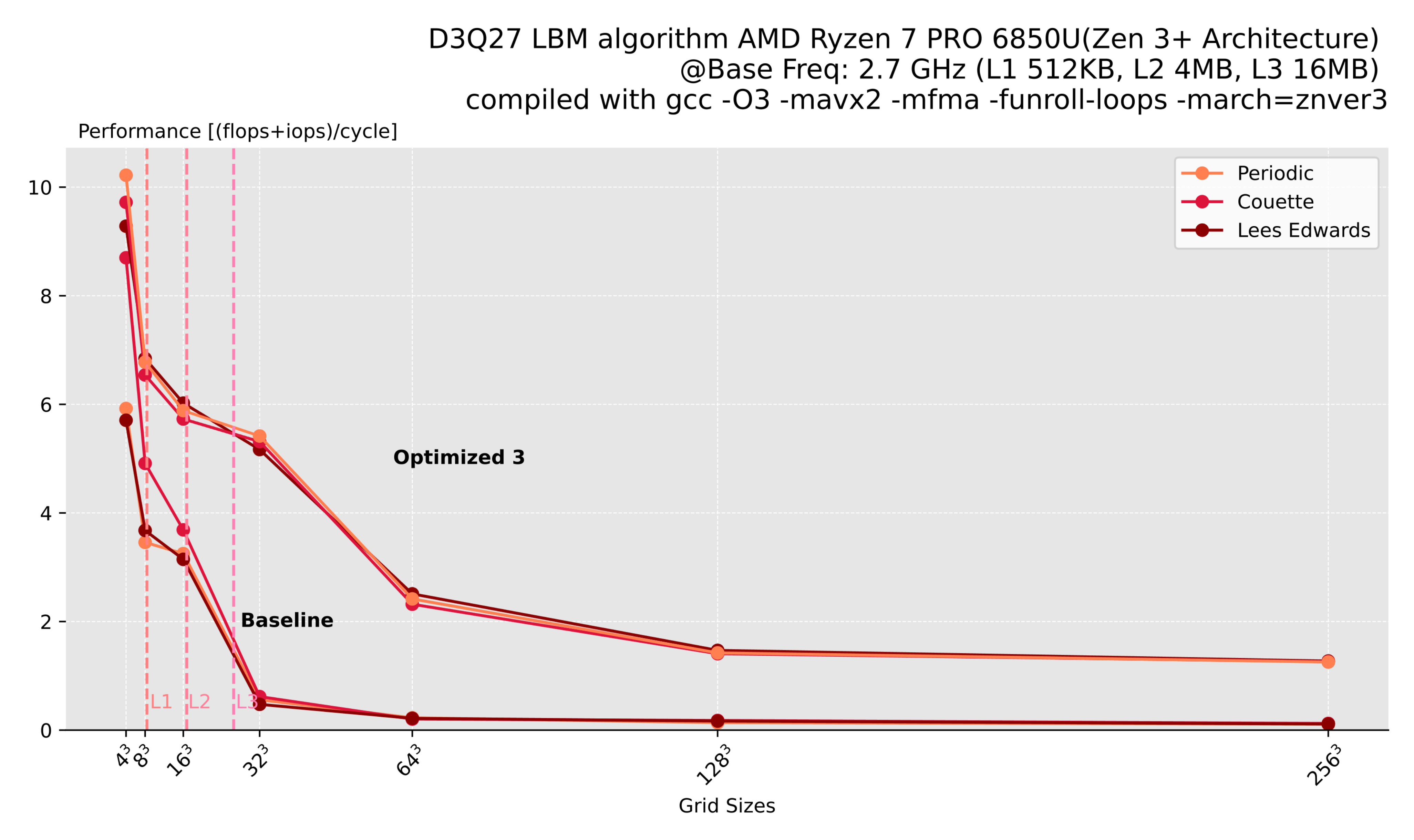
Performance – Baseline vs Optimization 1

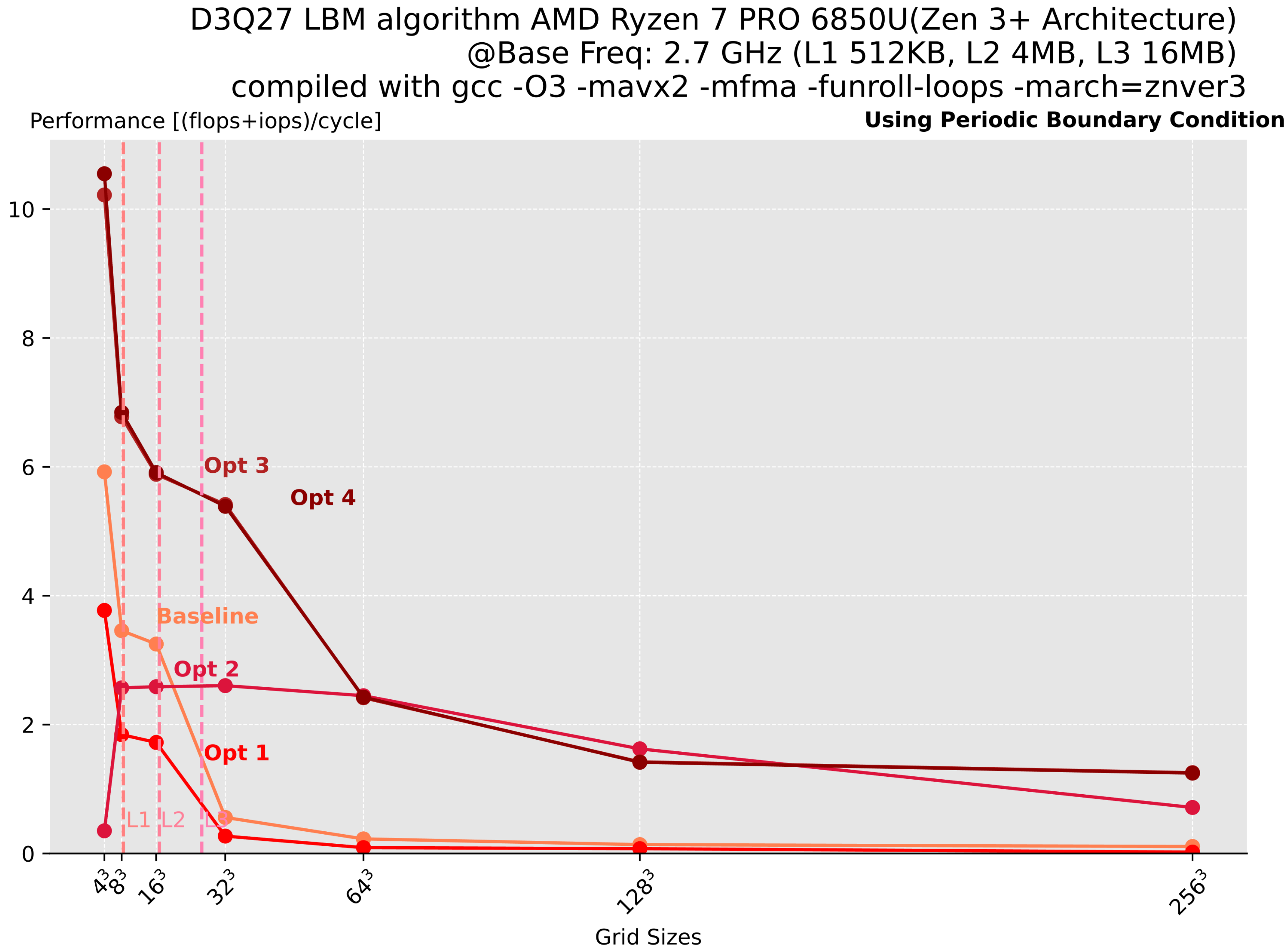


Performance – Baseline vs Optimization 2

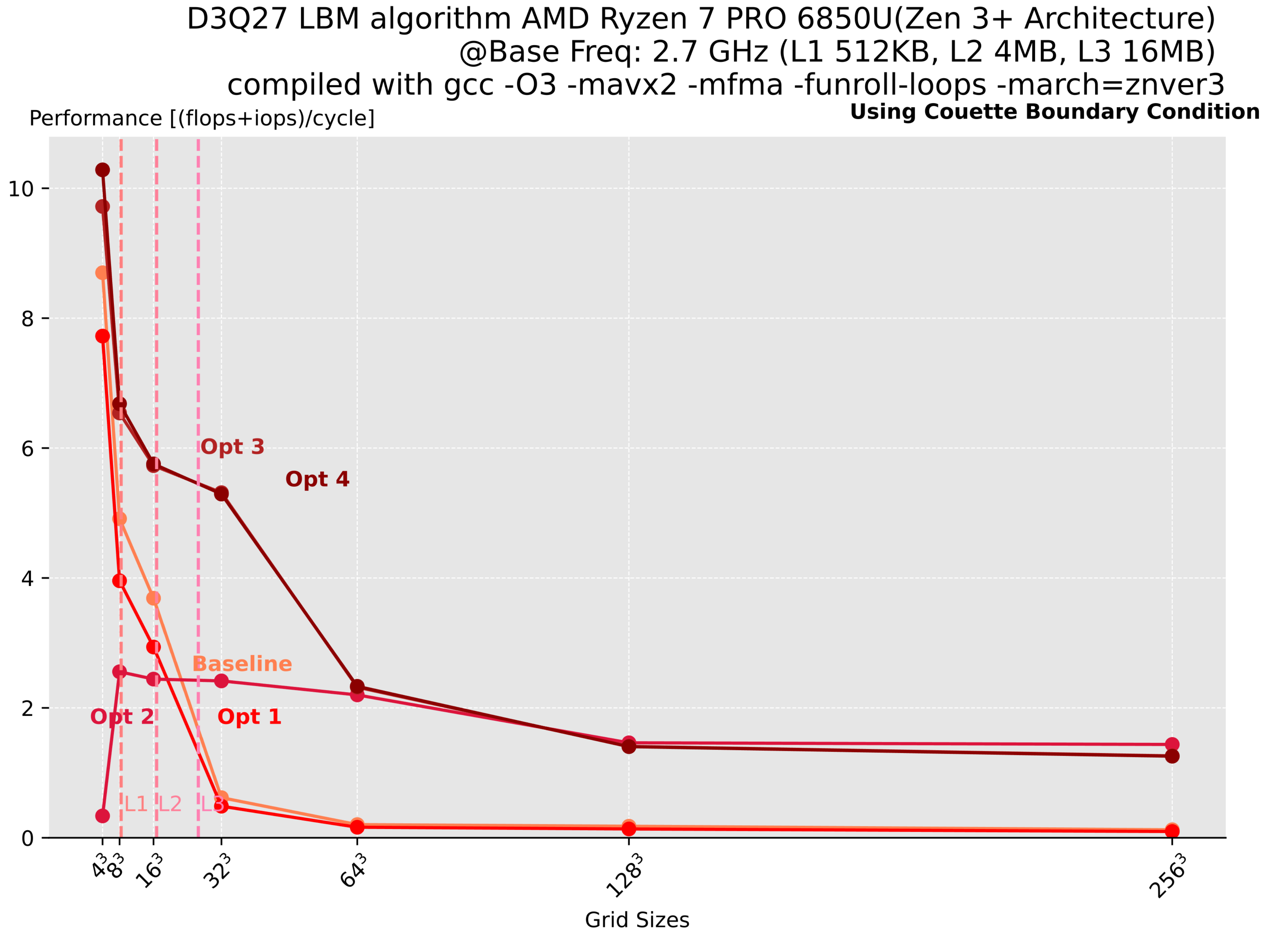


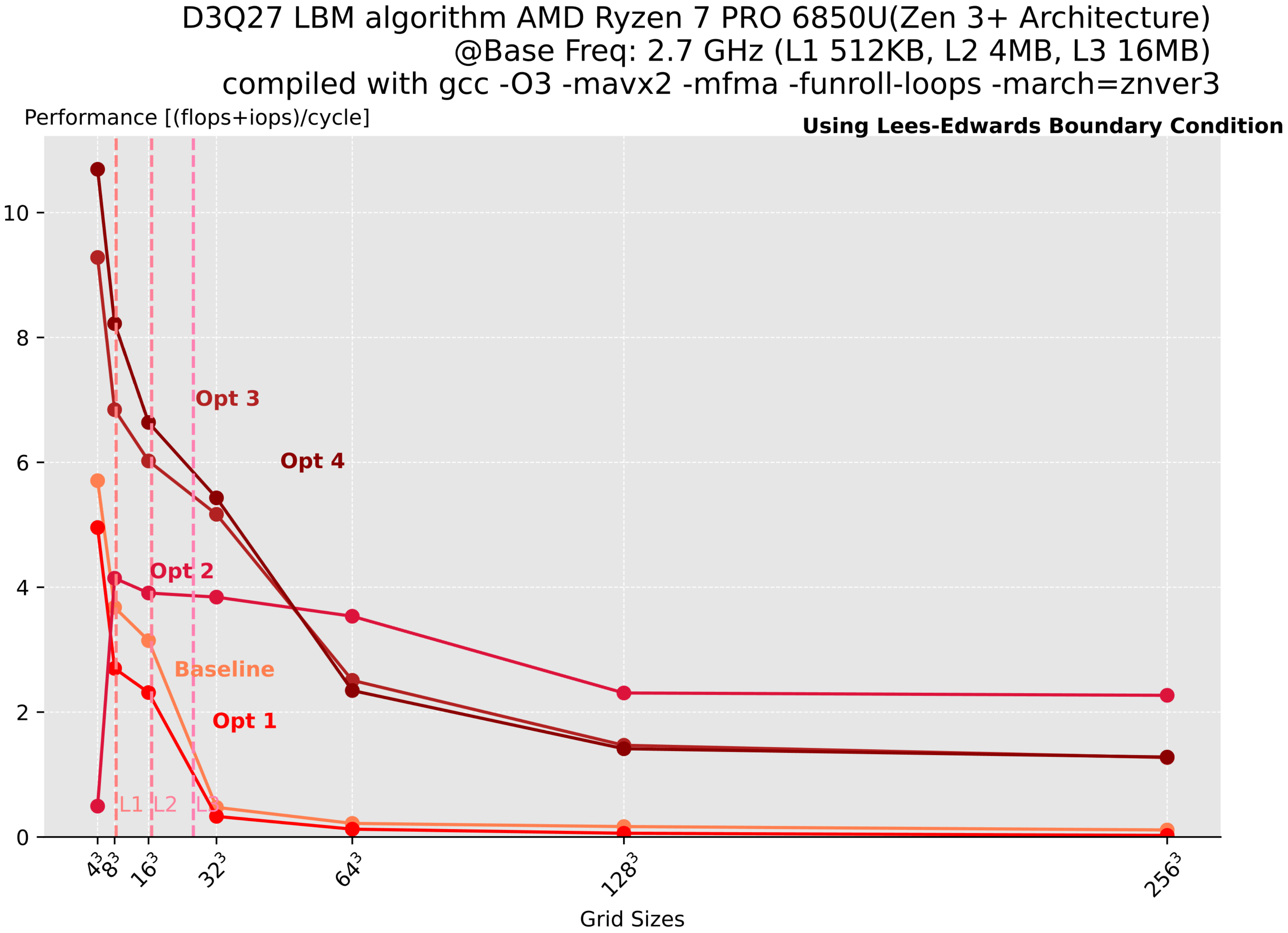
Performance – Baseline vs Optimization 3





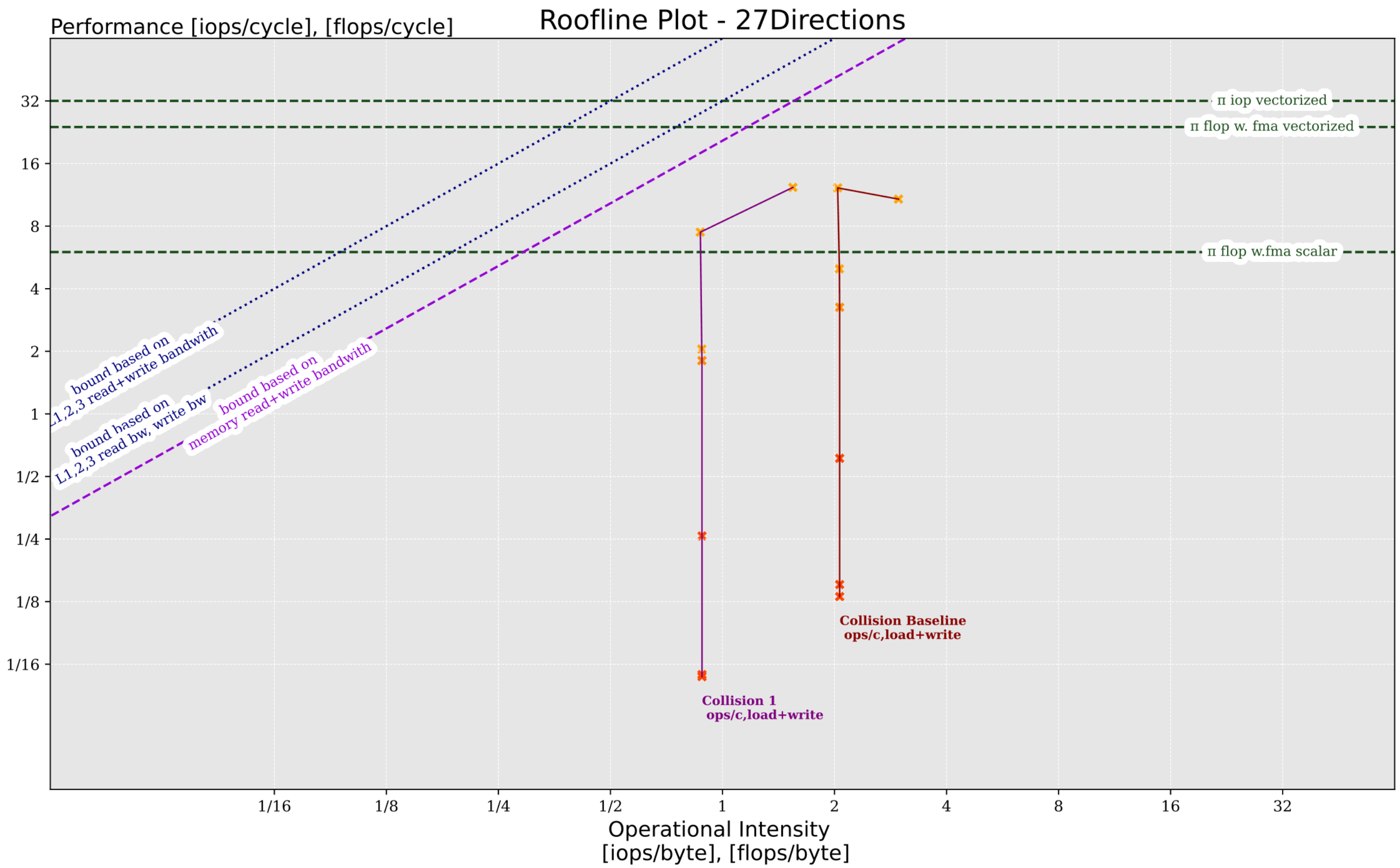
D3Q27 LBM algorithm AMD Ryzen 7 PRO 6850U(Zen 3+ Architecture)
@Base Freq: 2.7 GHz (L1 512KB, L2 4MB, L3 16MB)
compiled with gcc -O3 -mavx2 -mfma -funroll-loops -march=znver3
Performance [(flops+iops)/cycle]
Using Periodic Boundary Condition
Opt 3
Opt 4
Baseline
Opt 2
Opt 1
L1
L2
L3
0
2
4
6
8
10
4^3
8^3
16^3
32^3
64^3
128^3
256^3
Grid Sizes


D3Q27 LBM algorithm AMD Ryzen 7 PRO 6850U(Zen 3+ Architecture)
@Base Freq: 2.7 GHz (L1 512KB, L2 4MB, L3 16MB)
compiled with gcc -O3 -mavx2 -mfma -funroll-loops -march=znver3
Using Couette Boundary Condition
Performance [(flops+iops)/cycle]
Opt 3
Opt 4
Baseline
Opt 2
Opt 1
L1
L2
L3
0
2
4
6
8
10
4^3
8^3
16^3
32^3
64^3
128^3
256^3
Grid Sizes

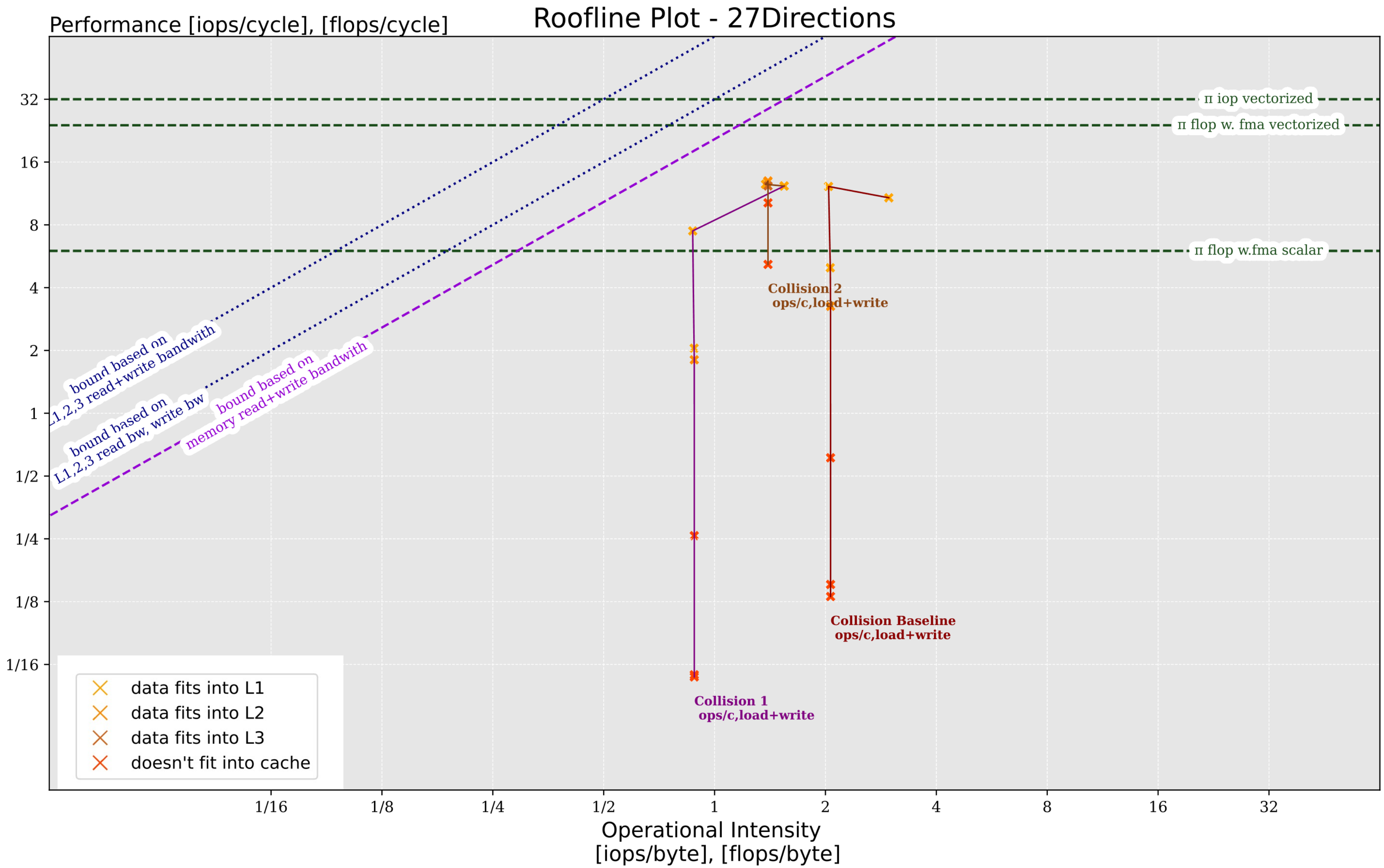

D3Q27 LBM algorithm AMD Ryzen 7 PRO 6850U(Zen 3+ Architecture)
@Base Freq: 2.7 GHz (L1 512KB, L2 4MB, L3 16MB)
compiled with gcc -O3 -mavx2 -mfma -funroll-loops -march=znver3
Performance [(flops+iops)/cycle]
Using Lees-Edwards Boundary Condition
Opt 3
Opt 4
Opt 2
Baseline
Opt 1
L1
L2
L3
0
2
4
6
8
10
4^3
8^3
16^3
32^3
64^3
128^3
256^3
Grid Sizes

Rooflines

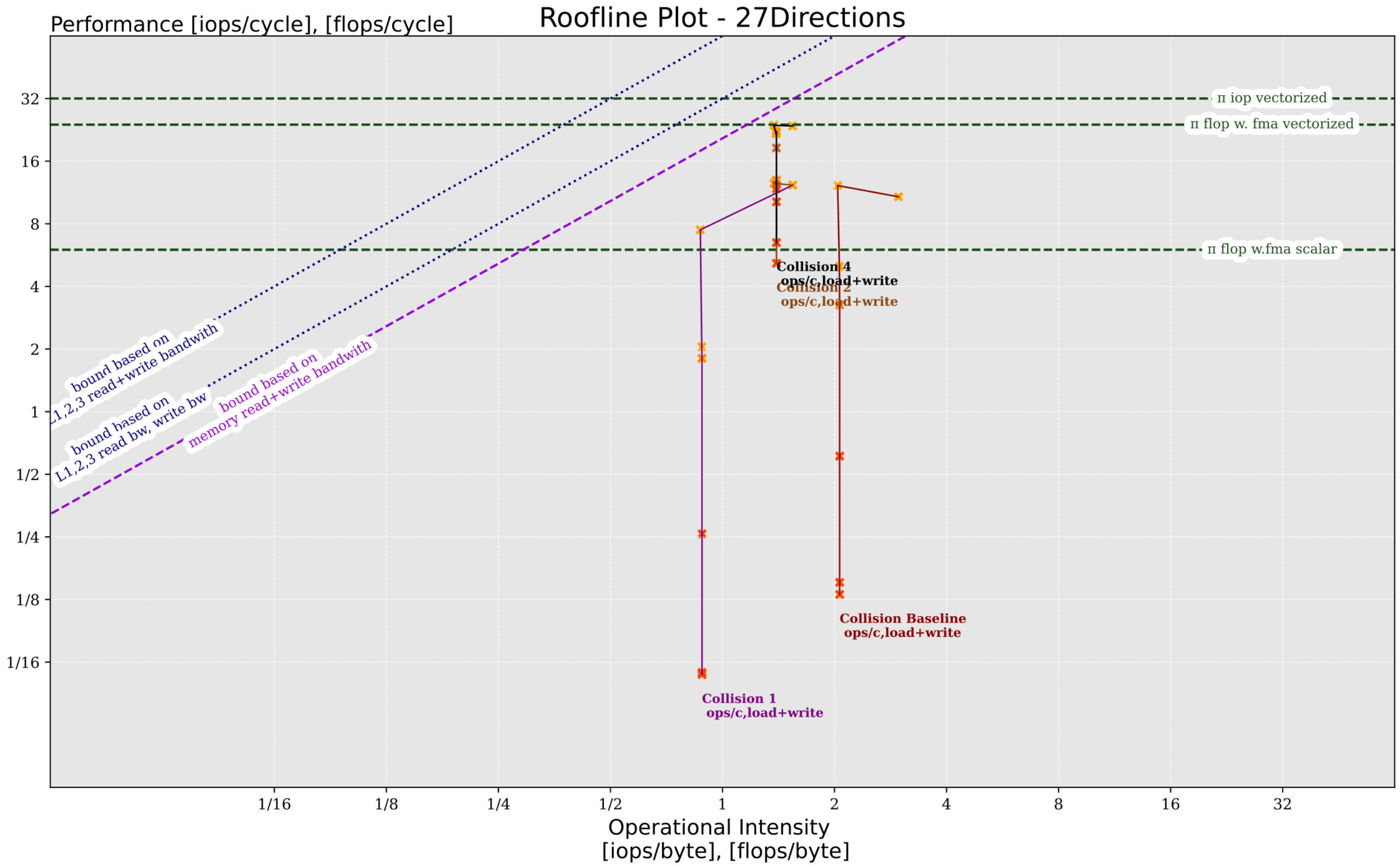
Collision 1


Roofline Plot - 27Directions
Performance [iops/cycle], [flops/cycle]
π iop vectorized
π flop w. fma vectorized
π flop w.fma scalar
bound based on
L1,2,3 read+write bandwith
bound based on
L1,2,3 read bw, write bw
bound based on
memory read+write bandwith
Collision Baseline
ops/c,load+write
Collision 1
ops/c,load+write
Operational Intensity
[iops/byte], [flops/byte]
32
16
8
4
2
1
1/2
1/4
1/8
1/16
1/16
1/8
1/4
1/2
1
2
4
8
16
32

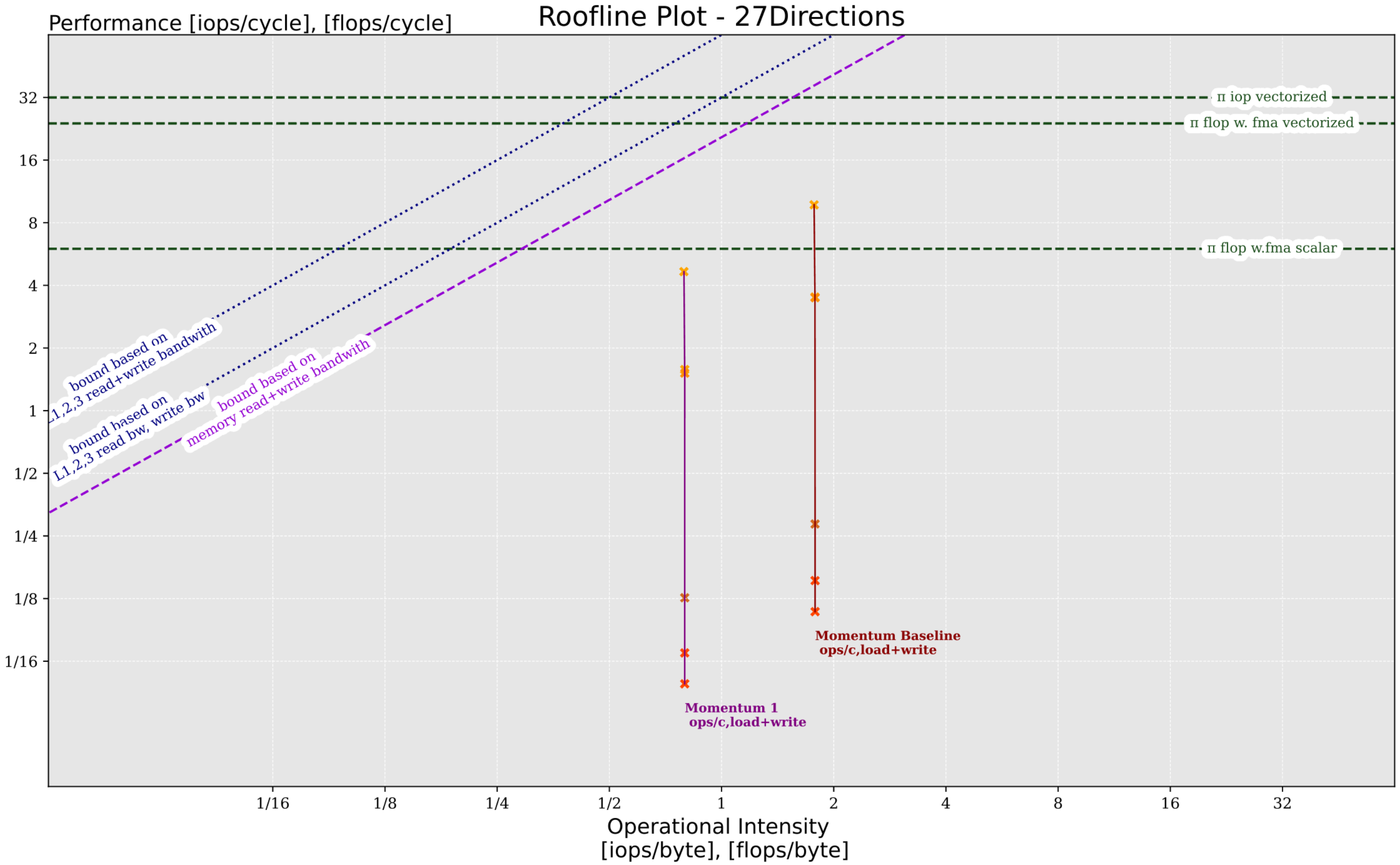
Optimization 2 - Collision



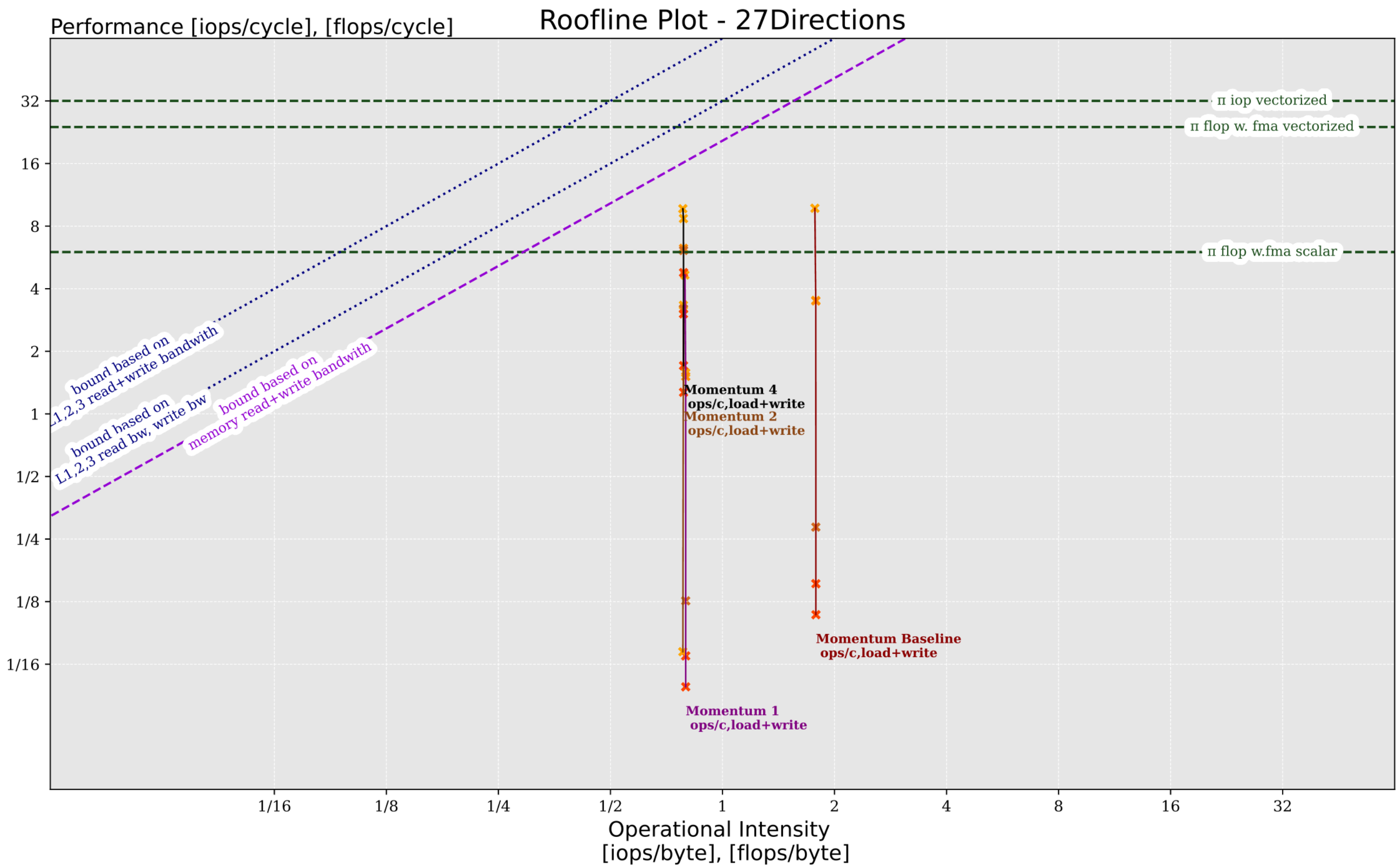
Collision 3 + 4



Momentum 1

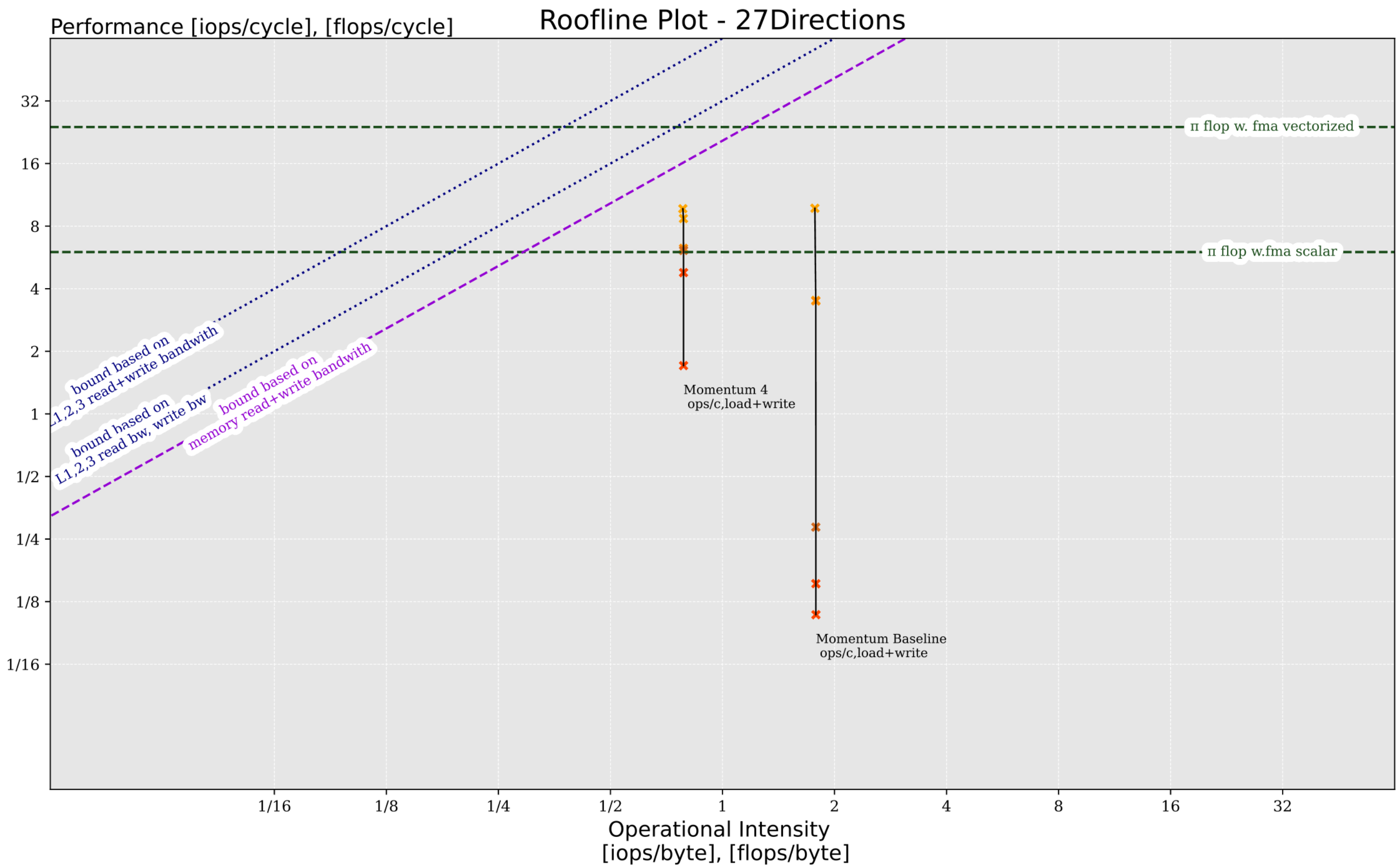

Roofline Plot - 27Directions
Performance [iops/cycle], [flops/cycle]
Operational Intensity
[iops/byte], [flops/byte]
π iop vectorized
π flop w. fma vectorized
π flop w.fma scalar
bound based on
L1,2,3 read+write bandwith
bound based on
L1,2,3 read bw, write bw
bound based on
memory read+write bandwith
Momentum Baseline
ops/c,load+write
Momentum 1
ops/c,load+write
32
16
8
4
2
1
1/2
1/4
1/8
1/16
1/16
1/8
1/4
1/2
1
2
4
8
16
32

Momentum 2 & 3



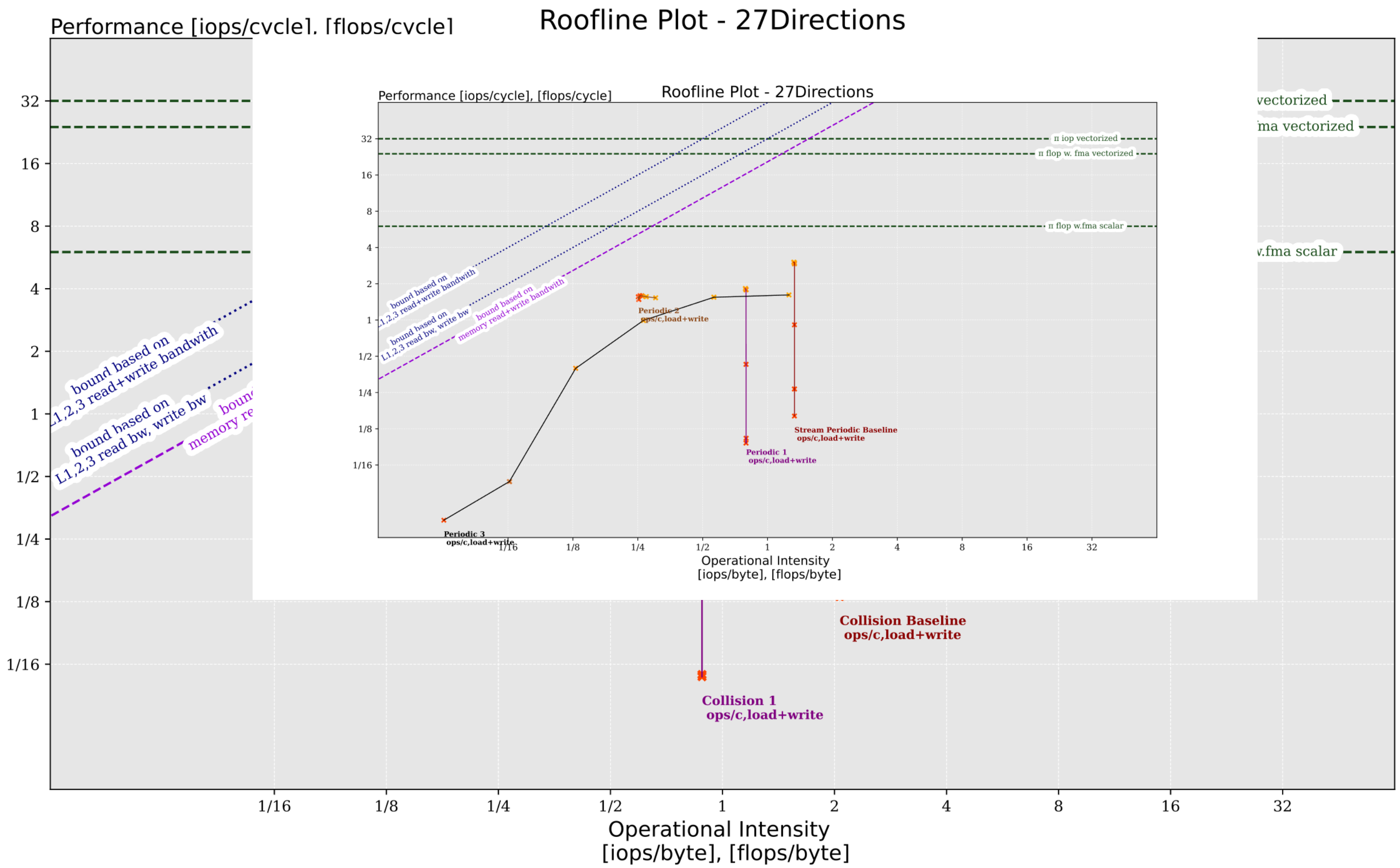
Roofline Plot - 27Directions
Performance [iops/cycle], [flops/cycle]
32
16
8
4
2
1
1/2
1/4
1/8
1/16
π iop vectorized
π flop w. fma vectorized
π flop w.fma scalar
bound based on
L1,2,3 read+write bandwith
bound based on
L1,2,3 read bw, write bw
bound based on
memory read+write bandwith
Momentum 4
ops/c,load+write
Momentum 2
ops/c,load+write
Momentum Baseline
ops/c,load+write
Momentum 1
ops/c,load+write
1/16
1/8
1/4
1/2
1
2
4
8
16
32
Operational Intensity
[iops/byte], [flops/byte]

Momentum 4



Roofline Plot - 27Directions
Performance [iops/cycle], [flops/cycle]
π flop w. fma vectorized
π flop w.fma scalar
bound based on L1,2,3 read+write bandwidth
bound based on L1,2,3 read bw, write bw
bound based on memory read+write bandwidth
Momentum 4 ops/c,load+write
Momentum Baseline ops/c,load+write
32
16
8
4
2
1
1/2
1/4
1/8
1/16
1/16
1/8
1/4
1/2
1
2
4
8
16
32
Operational Intensity [iops/byte], [flops/byte]

Without 256 rooflines



Boundary Conditions Comparison - Theory

Boundary Conditions

- **Periodic boundary conditions** are used to simulate an infinite system by replicating the simulation domain. When a particle **exits one side of the domain, it re-enters from the opposite side**.
- **Couette flow** describes the **flow between two parallel plates moving relative to each other**. Couette boundary conditions are used to model **shear-driven** flows.
- **Lees-Edwards boundary conditions** are an extension of periodic boundary conditions, used to **simulate shear flows in a periodic domain**. This involves adding a shear displacement to the particles when they cross the boundary.

Implementation of Boundary Conditions

- Periodic Boundary Conditions: Particles leaving one side of the domain are reintroduced on the opposite side, straightforward implementation due to regularity
- Couette Boundary Conditions: Boundary nodes are assigned velocities corresponding to the moving plates. Distribution functions at boundary nodes are updated to reflect the macroscopic velocities of the plates. The non-equilibrium bounce-back or halfway bounce-back schemes can be used to handle the moving boundaries.
- Lees-Edwards Boundary Conditions: Adjust particle positions by a shear displacement when they cross periodic boundaries. Ensures that the shear deformation is correctly applied to the particles crossing the boundary.

Comparison of the Boundary Conditions

- Periodic Boundary Conditions: Typically, the most computationally efficient due to their simplicity and regularity. High accuracy for bulk properties and homogeneous systems.
- Couette Boundary Conditions: Moderate computational cost due to additional calculations required for moving boundary nodes. Accurate for shear-driven flows.
- Lees-Edwards Boundary Conditions: Higher computational cost compared to periodic conditions due to shear displacement calculations and adjustments. Accurate for simulating shear flows in periodic domains.

END EXTRA SLIDES