Proxy or Imposter? A Method and Case Study to Determine the Answer

PRESENTED BY

Omar Aaziz (Sandia National Labs)

Jeanine Cook (Sandia National Labs)

Courtenay Vaughan (Sandia National Labs)

David Richards (Lawrence Livermore National Laboratory)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Proxy Apps are Wonderful!

Easy to Build!

Proxy Apps are Wonderful!

Easy to Modify!

Easy to Build!

Proxy Apps are Wonderful!

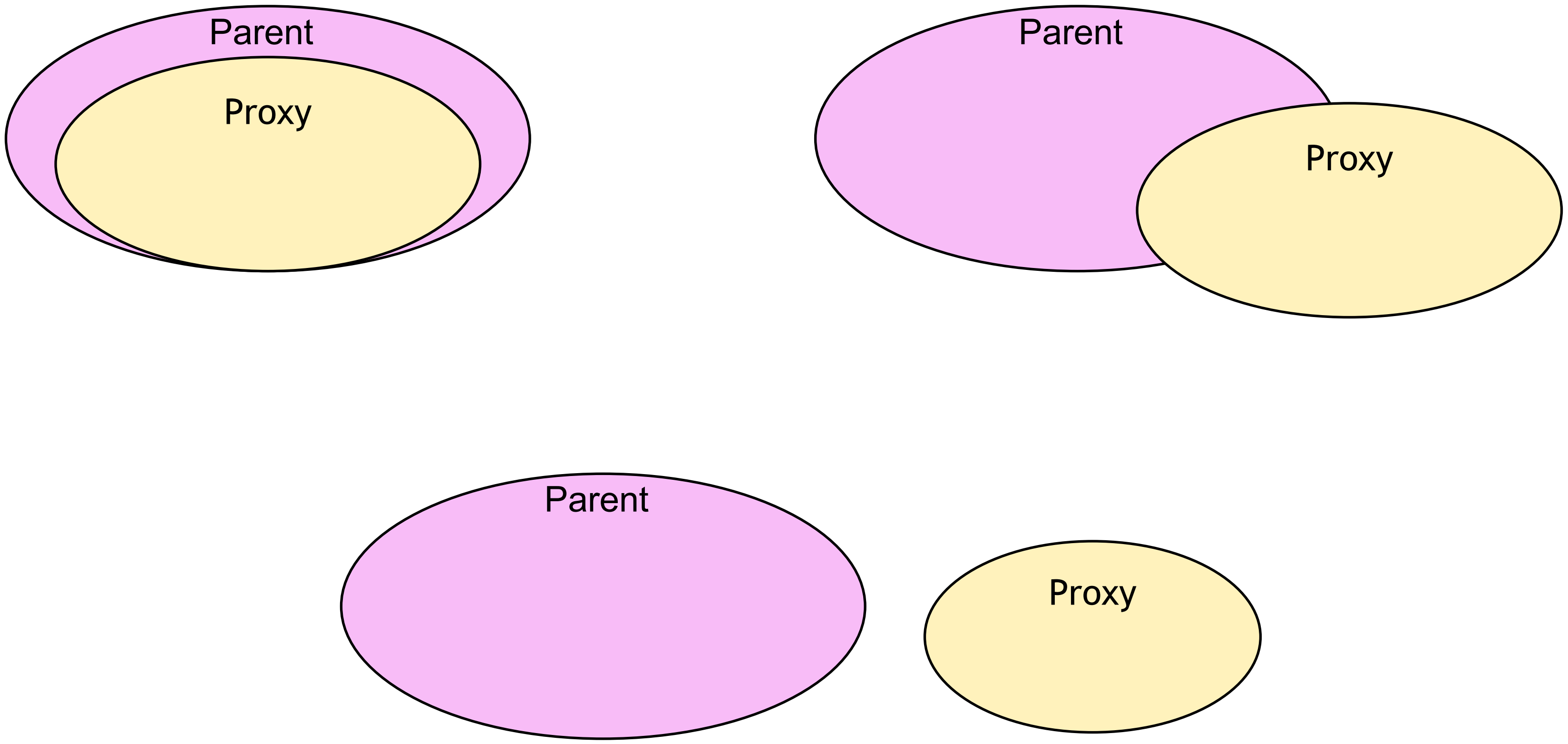
Easy to Modify!

Easy to Build!

Easy to Run!

Explore Ideas!

Do Proxies Match the Real Thing?



Dimensions: Resource Domains

Basic Node	Host processors and memory
Communication	Cluster interconnect
Accelerator	GPU, et al.
Storage I/O	Filesystem

Dimensions: Resource Domains

Basic Node	Host processors and memory
Commun	erconnect
Accelera	.
Storage I/O	Filesystem

Not much proxy support, or unavailable resources, so did not do

Dimensions: Resource Domains

Basic No	ssors and memory
Commun	erconnect
Accelerator	GPU, et al.
Storage I/O	Filesystem

Not much proxy support, or unavailable resources, so did not do

Dimensions: Resource Domains

Basic Node	Host processors and memory
Communication	Cluster interconnect
Accelerator	
Storage I/O	

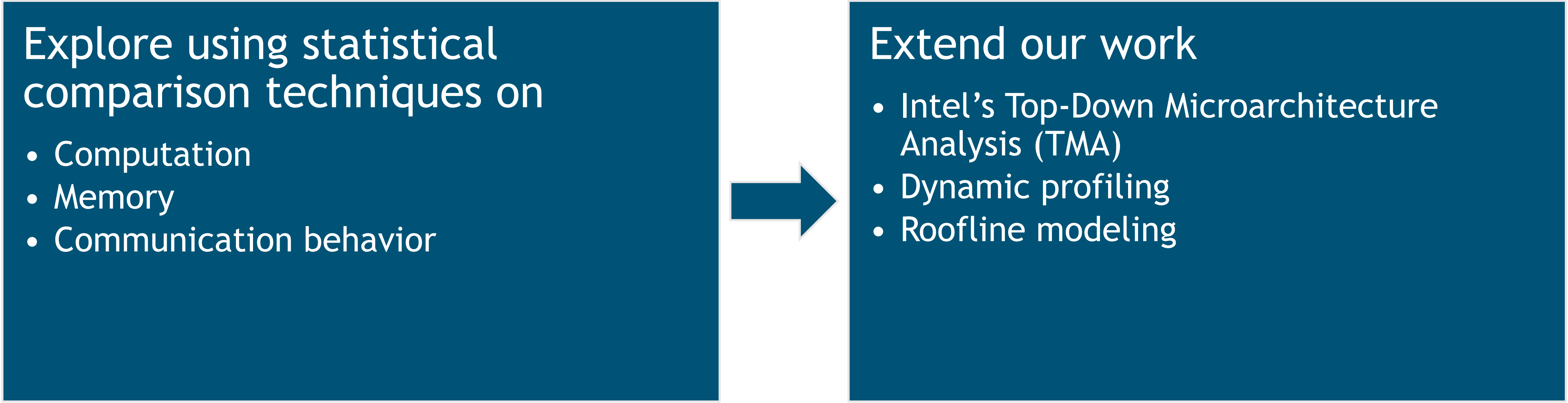
Success, but needs improvement

Dimensions: Resource Domains

Basic Node	Host processors and memory
Commun	erconnect
Accelera	.
Storage I/O	Filesystem

Good Result!

Target: Dynamic Behavior



Explore using statistical comparison techniques on

- Computation
- Memory
- Communication behavior

Extend our work

- Intel’s Top-Down Microarchitecture Analysis (TMA)
- Dynamic profiling
- Roofline modeling

Real and Proxy Applications

QMCPack

Quantum Monte Carlo code

material science to understand the electronic structure of molecular and solid state systems

Solve

Schrodinger wave equation

miniQMC

Quantum Monte Carlo code

comprises the important computational kernels of its parent

Real and Proxy Applications

QMCPack

Quantum Monte Carlo code

material science to understand the electronic structure of molecular and solid state systems

Solve

Schrodinger wave equation

miniQMC

Quantum Monte Carlo code

comprises the important computational [illegible] of its parent

Communication

Computational Platforms

INTEL SKYLAKE PLATINUM 8160

24 CORES

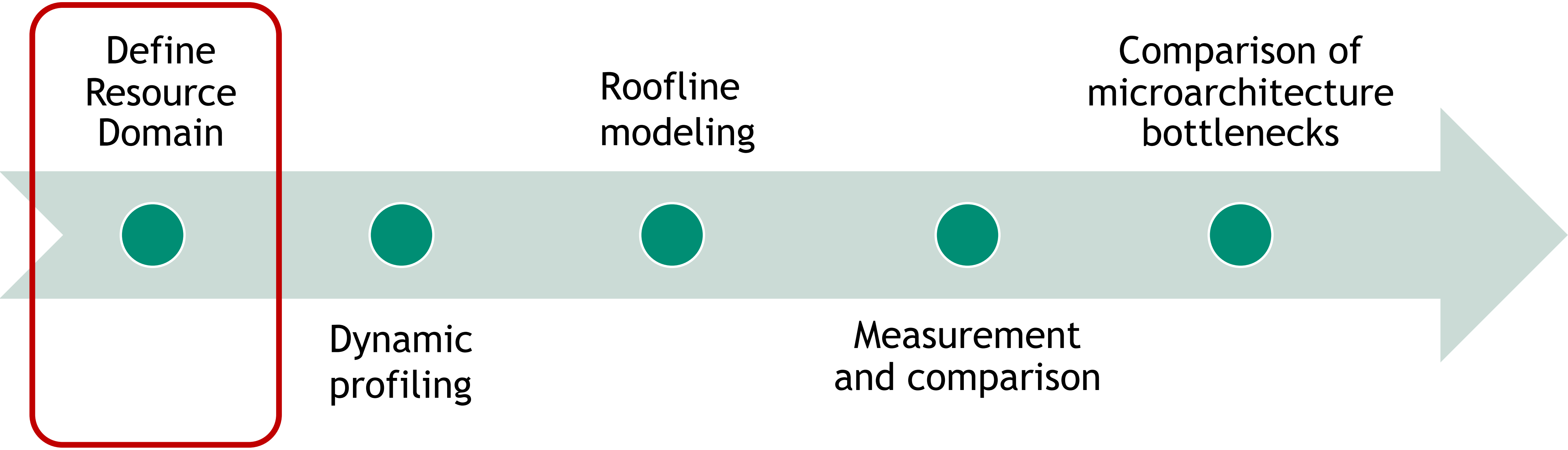
6 MEMORY CHANNELS PER SOCKET

SUPPORTS THE AVX512

TOTAL NODES 40

INTERCONNECT, INTEL OMNIPATH

Methodology



Define Resource Domain

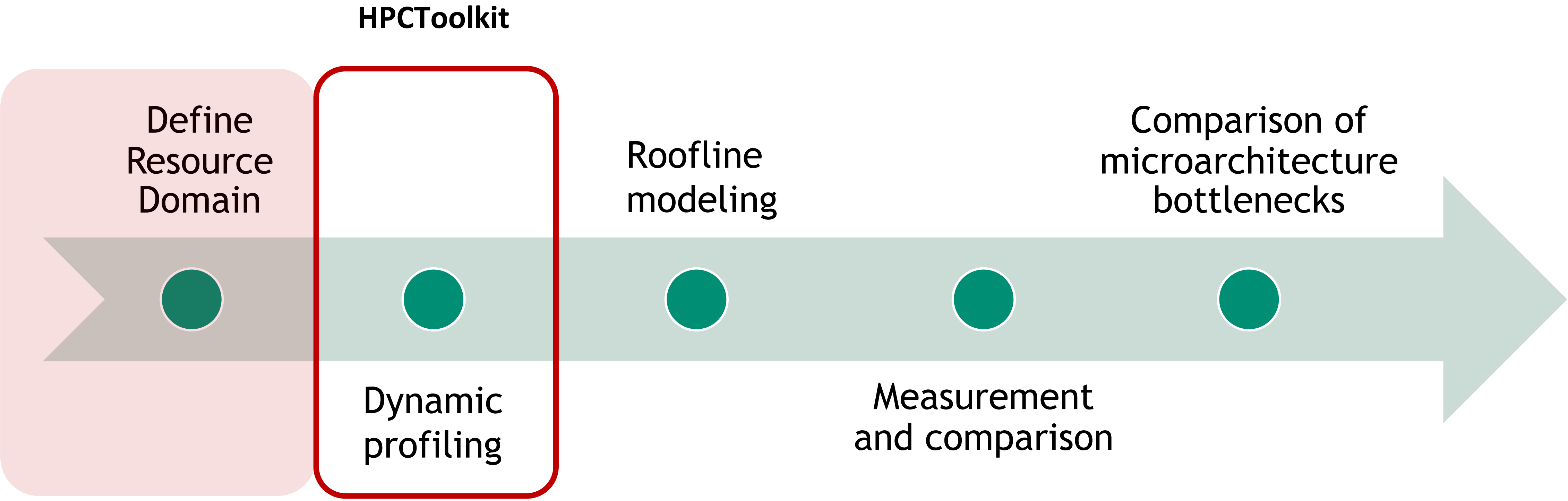
Dynamic profiling

Roofline modeling

Measurement and comparison

Comparison of microarchitecture bottlenecks

Methodology



HPCToolkit

Define Resource Domain

Dynamic profiling

Roofline modeling

Measurement and comparison

Comparison of microarchitecture bottlenecks

Dynamic profiling

Understand if key **kernels** and **functions** implementing these kernels are **consistent** across the two applications

kernels	Determinant update	Computationally intensive kernel
	Splines - SPO	Memory intensive kernel
	Jastrow factors	Computationally intensive kernel
	Distance tables	Memory intensive kernel

Kernel Percentages of Execution Time

Kernel	miniQMC	Time %	QMCPACK	Time %
Determinant	DiracDeterminant::acceptMove DiracDeterminant::ratioGrad MKL DiracDeterminant::ratio	57.8 6.2 5.2 4.7	DiracDeterminantBase::acceptMove DiracDeterminantBase::ratioGrad DiracDeterminantBase::ratio MKL DiracDeterminantBase::evaluateLog	49.3 7.7 2.3 10.0 2.3
Single-Particle Orbital (SPO)	einspline_spo::MultiBspline::evaluate_vgh einspline_spo::MultiBspline::evaluate_v einspline_spo::MultiBspline::set	9.3 1.2 1.0	SPOSetBuilderFactory::createSPOSet	11.0
Distance	ParticleSet::makeMoveAndCheck ParticleSet::setActive DistanceTableAA::makeMoveOnSphere	4.8 4.8 3.2	ParticleSet::makeMoveOnSphere ParticleSet::makeMoveAndCheck	11.2 1.0
Two Body Jastrow	TwoBodyJastrowOrbital::BsplineFunctor::acceptMove	1.4	TwoBodyJastrowOrbital::BsplineFunctor::ratio OneBodyJastrowOrbital::BsplineFunctor::ratioGrad	4.0 0.5

Kernel Percentages of Execution Time

Kernel	miniQMC	Time %	QMCPACK	Time %
Determinant	DiracDeterminant::acceptMove DiracDeterminant::ratioGrad MKL DiracDeterminant::ratio	57.8 6.2 5.2 4.7	DiracDeterminantBase::acceptMove DiracDeterminantBase::ratioGrad DiracDeterminantBase::ratio MKL DiracDeterminantBase::evaluateLog	49.3 7.7 2.3 10.0 2.3
Single-Particle Orbital (SPO)	einspline_spo::MultiBspline::evaluate_vgh einspline_spo::MultiBspline::evaluate_v einspline_spo::MultiBspline::set	9.3 1.2 1.0	SPOSetBuilderFactory::createSPOSet	11.0
Distance	ParticleSet::makeMoveAndCheck ParticleSet::setActive DistanceTableAA::makeMoveOnSphere	4.8 4.8 3.2	ParticleSet::makeMoveOnSphere ParticleSet::makeMoveAndCheck	11.2 1.0
Two Body Jastrow	TwoBodyJastrowOrbital::BsplineFunctor::acceptMove	1.4	TwoBodyJastrowOrbital::BsplineFunctor::ratio OneBodyJastrowOrbital::BsplineFunctor::ratioGrad	4.0 0.5

Kernel Percentages of Execution Time

Kernel	miniQMC	Time %	QMCPACK	Time %
Determinant	DiracDeterminant::acceptMove DiracDeterminant::ratioGrad MKL DiracDeterminant::ratio	57.8 6.2 5.2 4.7	DiracDeterminantBase::acceptMove DiracDeterminantBase::ratioGrad DiracDeterminantBase::ratio MKL DiracDeterminantBase::evaluateLog	49.3 7.7 2.3 10.0 2.3
Single-Particle Orbital (SPO)	einspline_spo::MultiBspline::evaluate_vgh einspline_spo::MultiBspline::evaluate_v einspline_spo::MultiBspline::set	9.3 1.2 1.0	SPOSetBuilderFactory::createSPOSet	11.0
Distance	ParticleSet::makeMoveAndCheck ParticleSet::setActive DistanceTableAA::makeMoveOnSphere	4.8 4.8 3.2	ParticleSet::makeMoveOnSphere ParticleSet::makeMoveAndCheck	11.2 1.0
Two Body Jastrow	TwoBodyJastrowOrbital::BsplineFunctor::acceptMove	1.4	TwoBodyJastrowOrbital::BsplineFunctor::ratio OneBodyJastrowOrbital::BsplineFunctor::ratioGrad	4.0 0.5

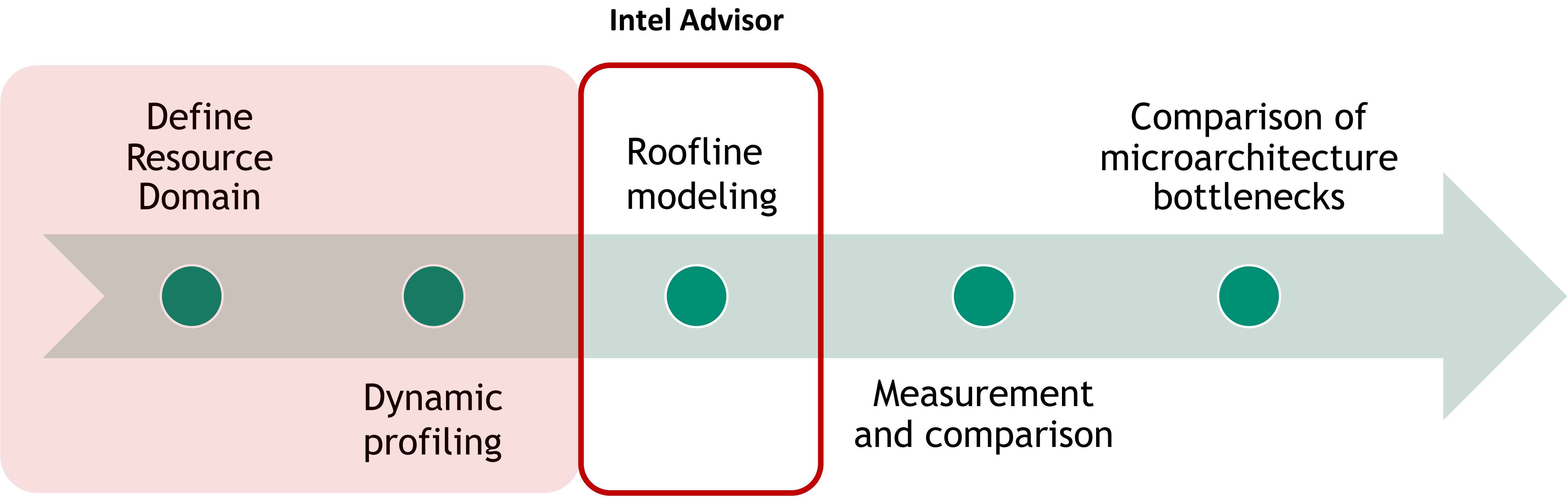
Kernel Percentages of Execution Time

Kernel	miniQMC	Time %	QMCPACK	Time %
Determinant	DiracDeterminant::acceptMove DiracDeterminant::ratioGrad MKL DiracDeterminant::ratio	57.8 6.2 5.2 4.7	DiracDeterminantBase::acceptMove DiracDeterminantBase::ratioGrad DiracDeterminantBase::ratio MKL DiracDeterminantBase::evaluateLog	49.3 7.7 2.3 10.0 2.3
Single-Particle Orbital (SPO)	einspline_spo::MultiBspline::evaluate_vgh einspline_spo::MultiBspline::evaluate_v einspline_spo::MultiBspline::set	9.3 1.2 1.0	SPOSetBuilderFactory::createSPOSet	11.0
Distance	ParticleSet::makeMoveAndCheck ParticleSet::setActive DistanceTableAA::makeMoveOnSphere	4.8 4.8 3.2	ParticleSet::makeMoveOnSphere ParticleSet::makeMoveAndCheck	11.2 1.0
Two Body Jastrow	TwoBodyJastrowOrbital::BsplineFunctor::acceptMove	1.4	TwoBodyJastrowOrbital::BsplineFunctor::ratio OneBodyJastrowOrbital::BsplineFunctor::ratioGrad	4.0 0.5

Kernel Percentages of Execution Time

Kernel	miniQMC	Time %	QMCPACK	Time %
Determinant	DiracDeterminant::acceptMove DiracDeterminant::ratioGrad MKL DiracDeterminant::ratio	57.8 6.2 5.2 4.7	DiracDeterminantBase::acceptMove DiracDeterminantBase::ratioGrad DiracDeterminantBase::ratio MKL DiracDeterminantBase::evaluateLog	49.3 7.7 2.3 10.0 2.3
Single-Particle Orbital (SPO)	einspline_spo::MultiBspline::evaluate_vgh einspline_spo::MultiBspline::evaluate_v einspline_spo::MultiBspline::set	9.3 1.2 1.0	SPOSetBuilderFactory::createSPOSet	11.0
Distance	ParticleSet::makeMoveAndCheck ParticleSet::setActive DistanceTableAA::makeMoveOnSphere	4.8 4.8 3.2	ParticleSet::makeMoveOnSphere ParticleSet::makeMoveAndCheck	11.2 1.0
Two Body Jastrow	TwoBodyJastrowOrbital::BsplineFunctor::acceptMove	1.4	TwoBodyJastrowOrbital::BsplineFunctor::ratio OneBodyJastrowOrbital::BsplineFunctor::ratioGrad	4.0 0.5

Methodology



Intel Advisor

Define Resource Domain

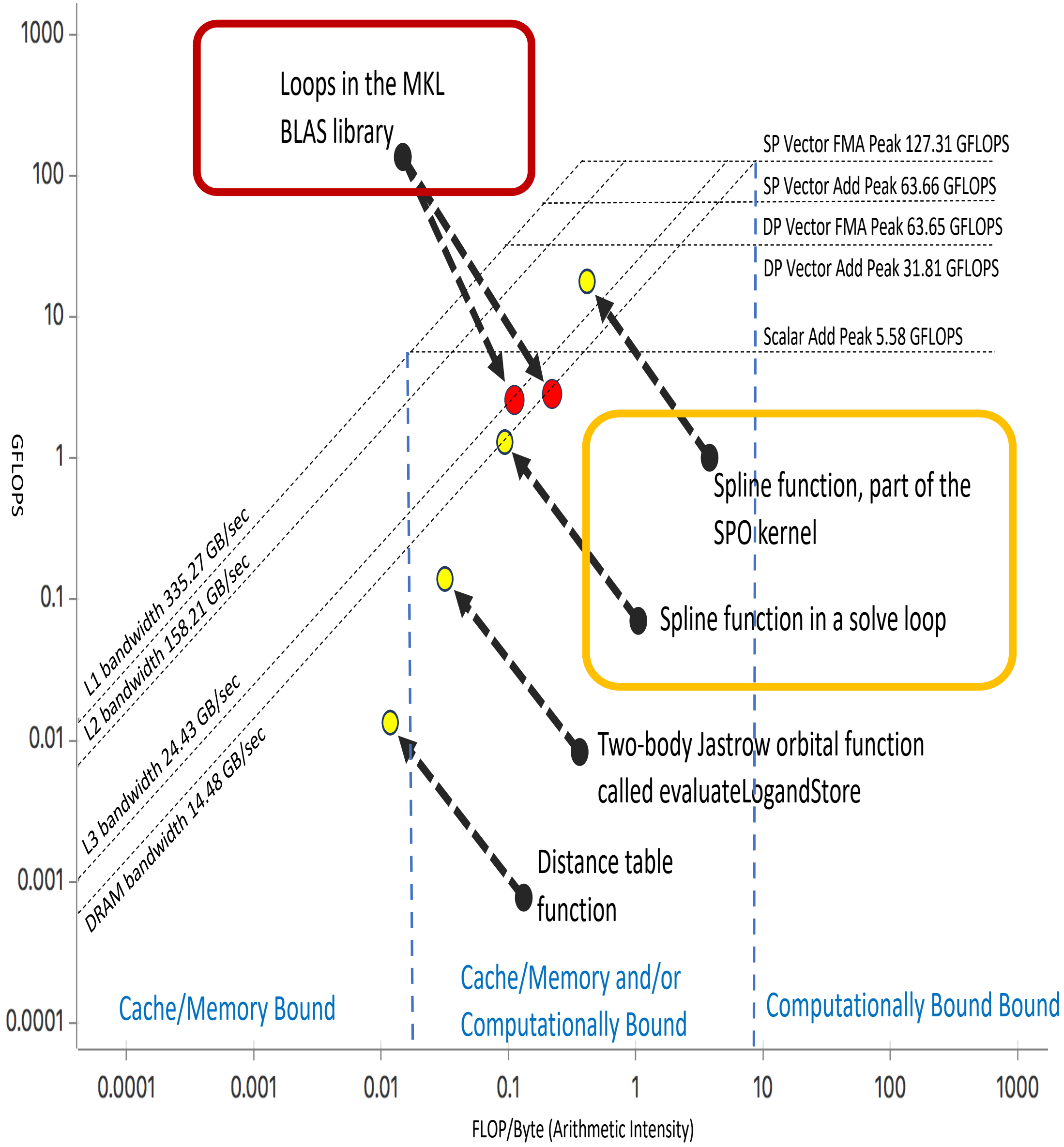
Dynamic profiling

Roofline modeling

Measurement and comparison

Comparison of microarchitecture bottlenecks

Roofline Model



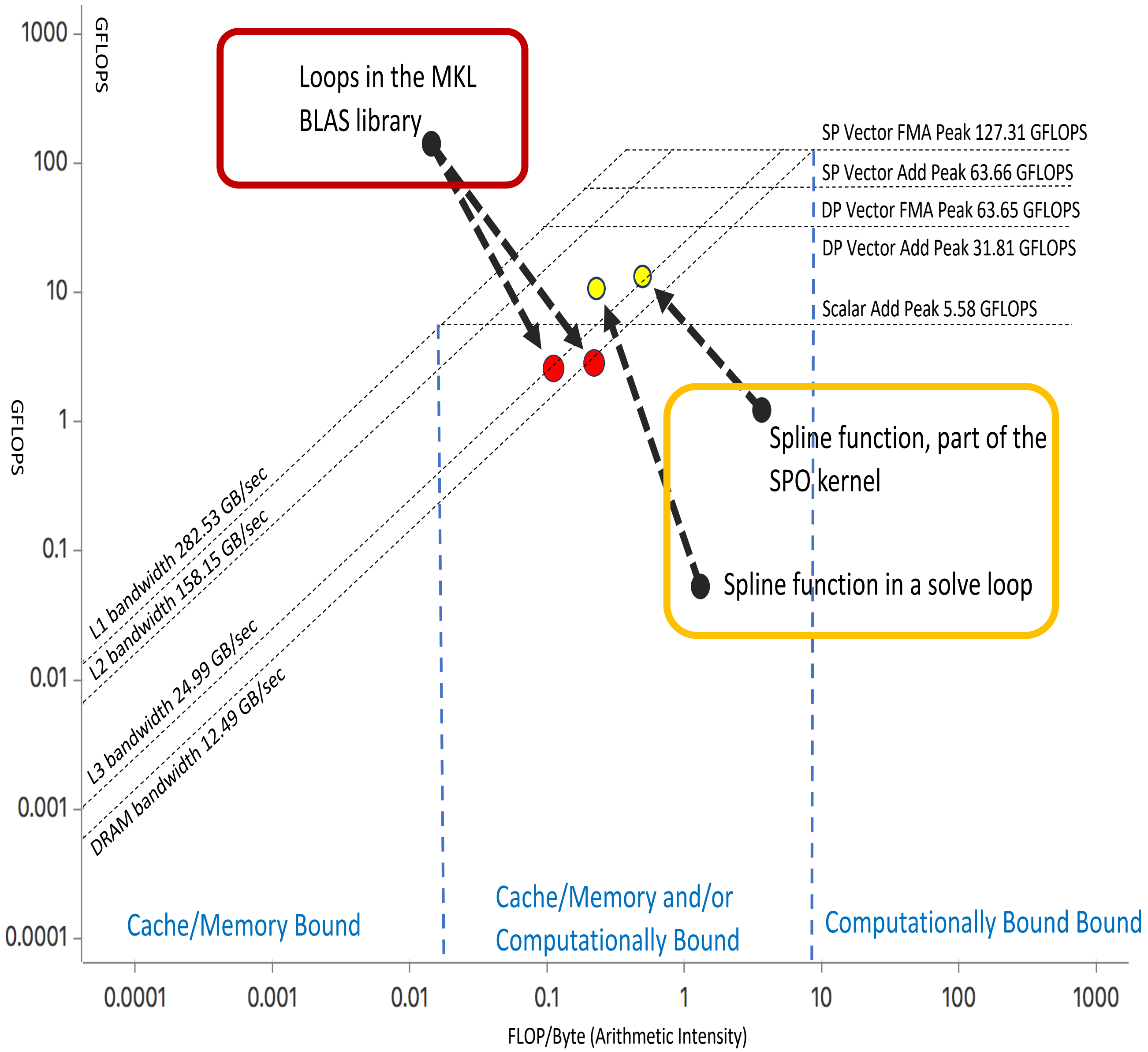
Loops in the MKL BLAS library

Spline function, part of the SPO kernel

Spline function in a solve loop

Two-body Jastrow orbital function called evaluateLogandStore

Distance table function

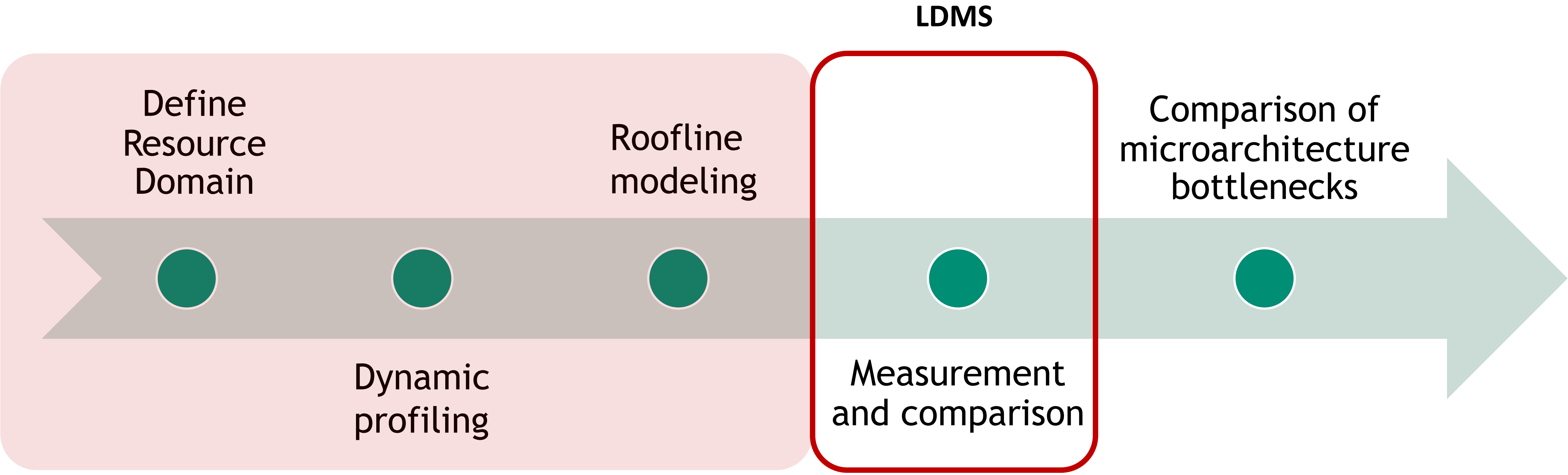


Loops in the MKL BLAS library

Spline function, part of the SPO kernel

Spline function in a solve loop

Methodology


LDMS
Define Resource Domain
Roofline modeling
Comparison of microarchitecture bottlenecks
Dynamic profiling
Measurement and comparison

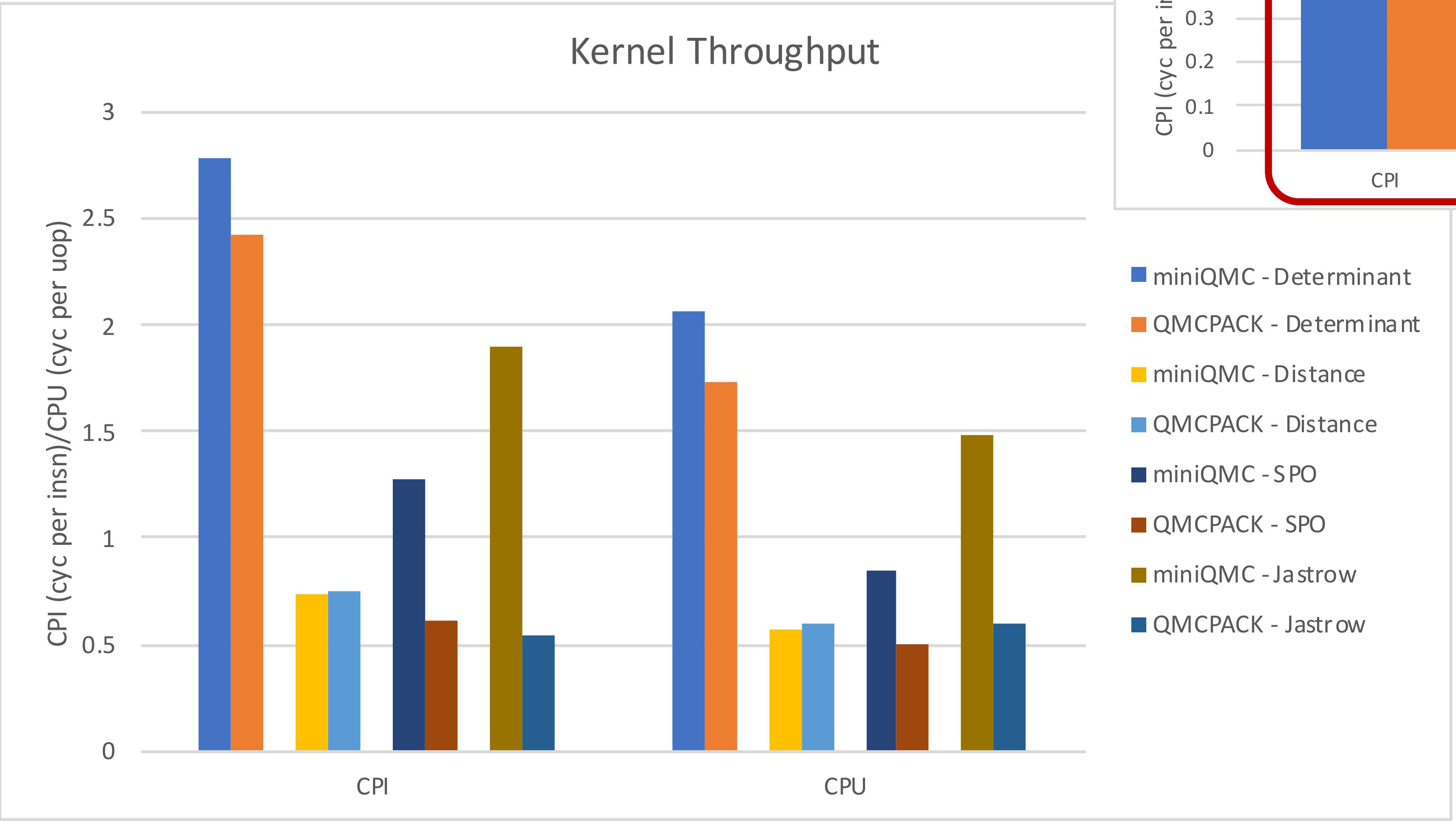
Understand Behavior Similarity

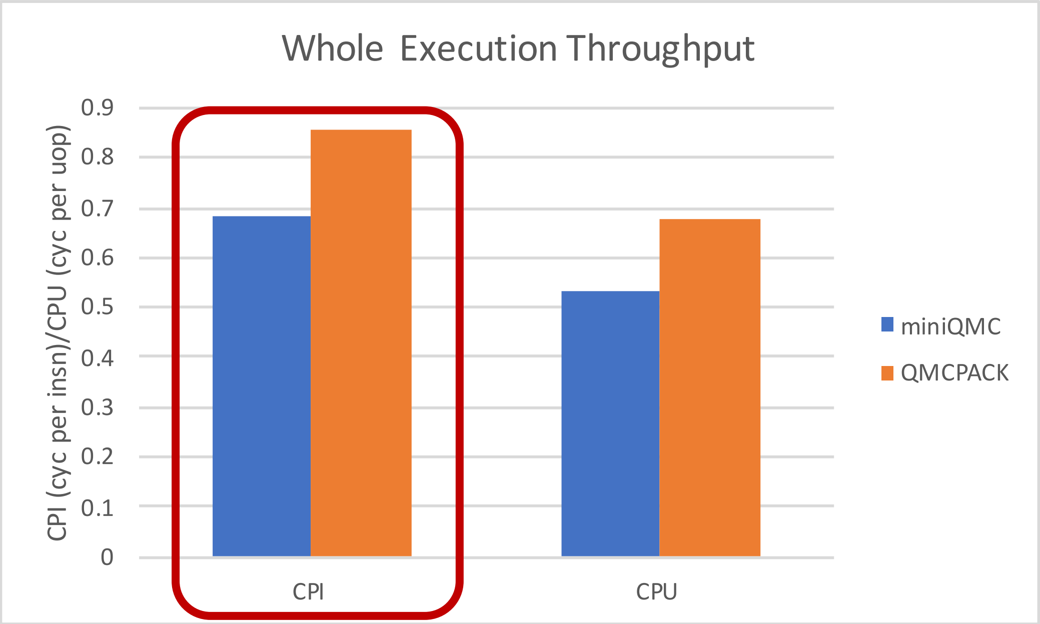
Events are derived for Likwid, with more

Average per-core

Derived from events measured during the whole execution of each application.

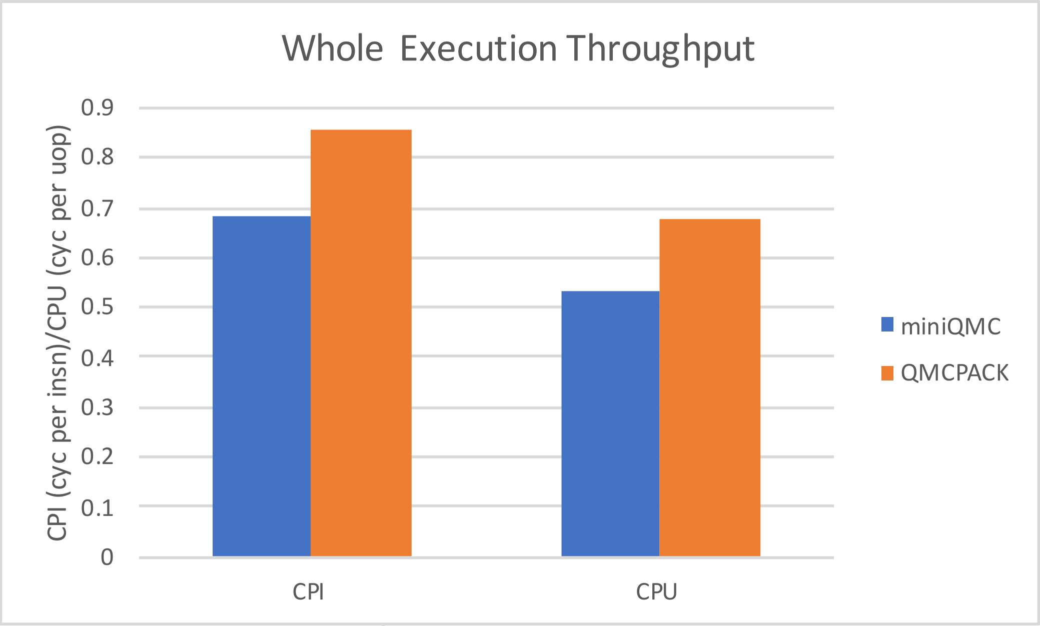
Understand Behavior Similarity

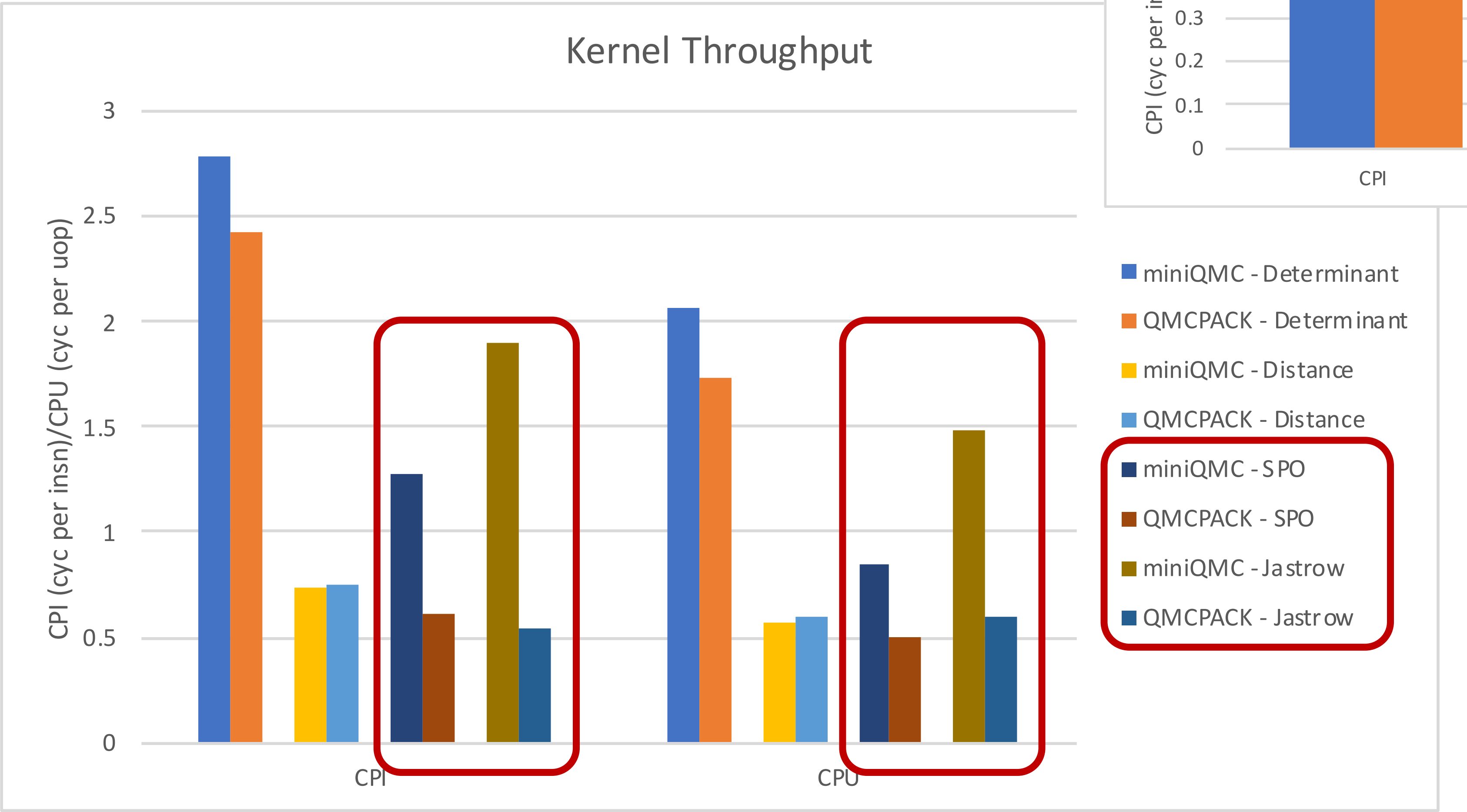




miniQMC
higher throughput

Understand Behavior Similarity





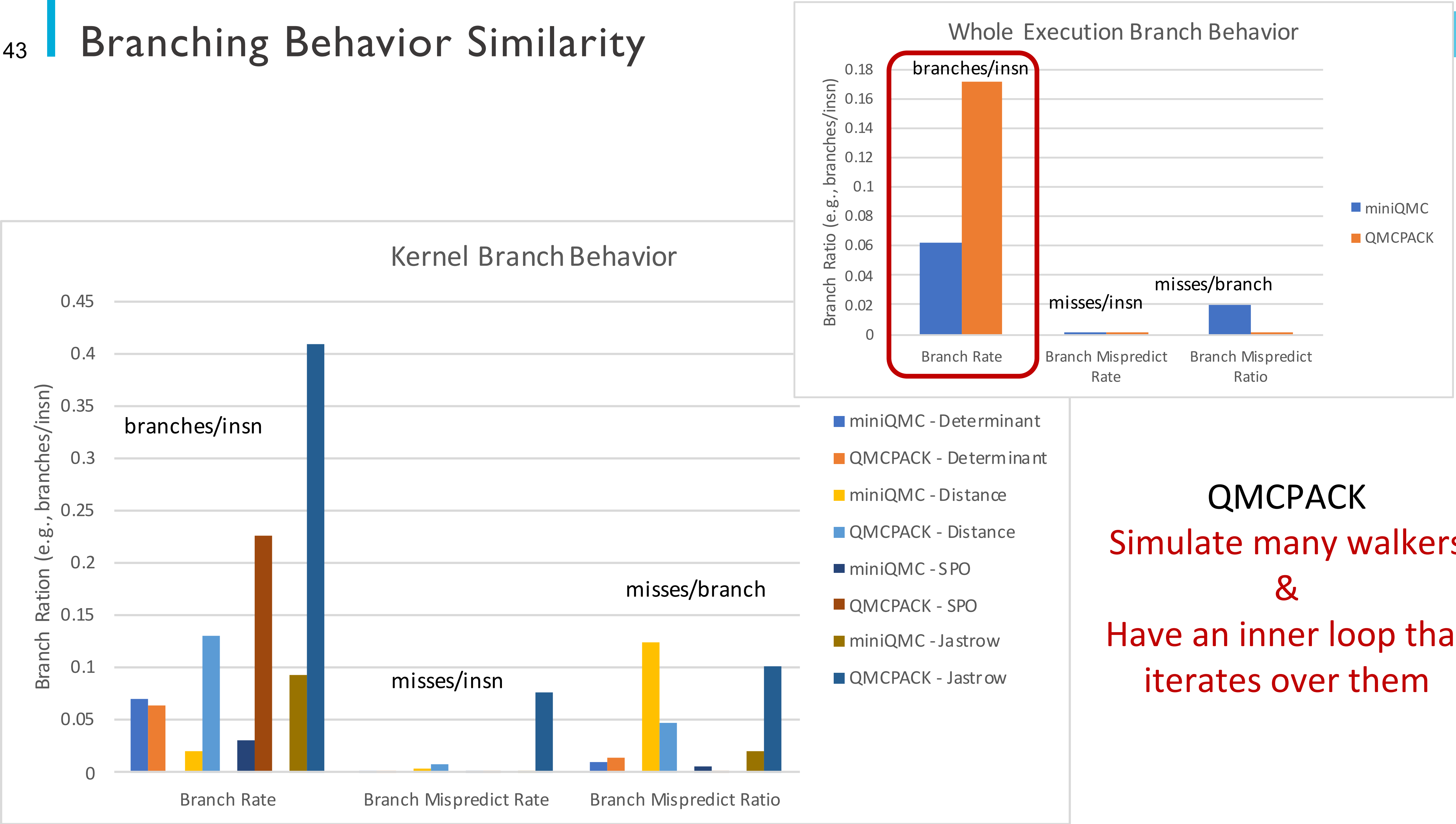
Jastrow kernel

&

SPO

no function similarity

Branching Behavior Similarity



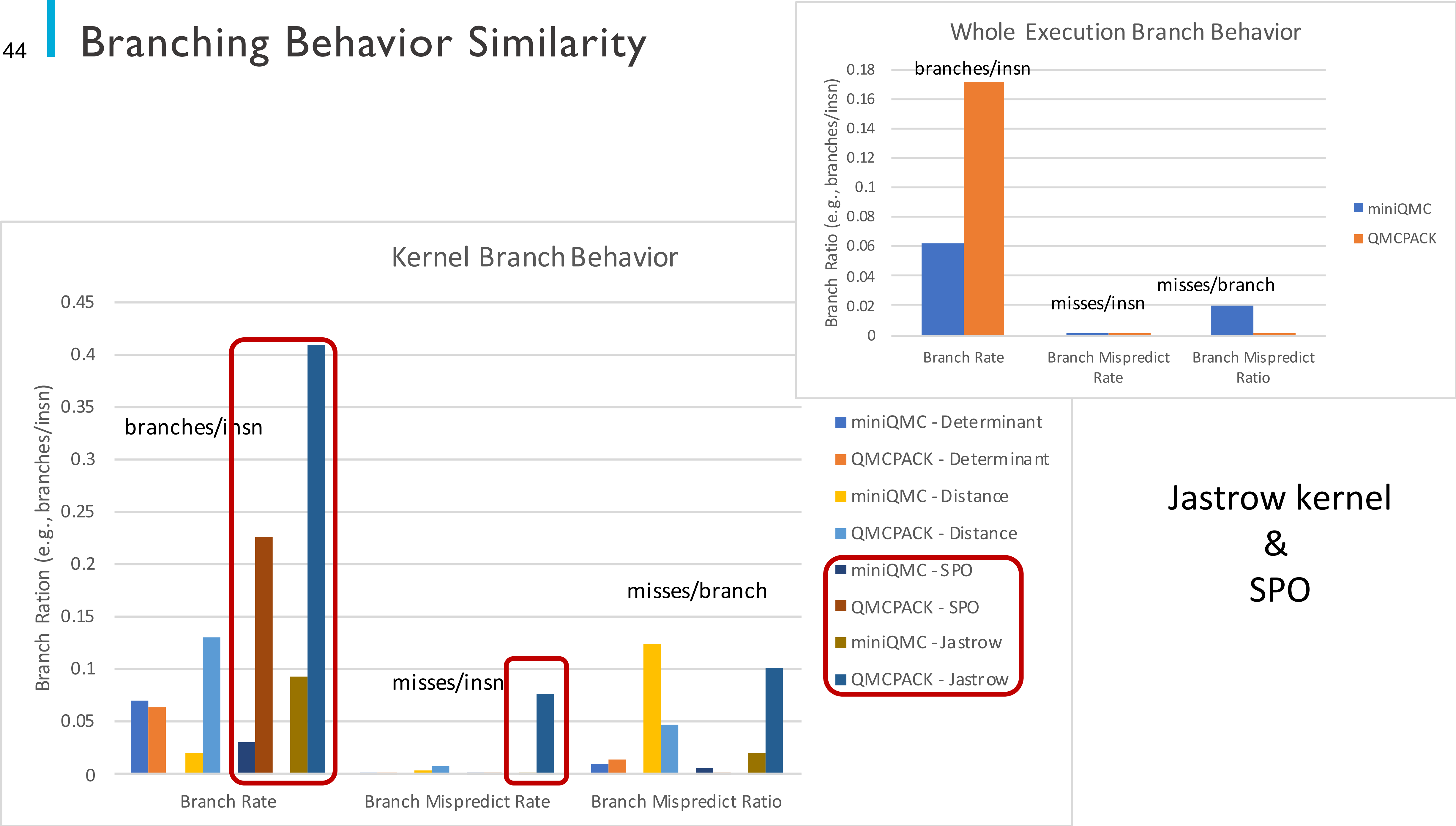
QMCPACK

Simulate many walkers

&

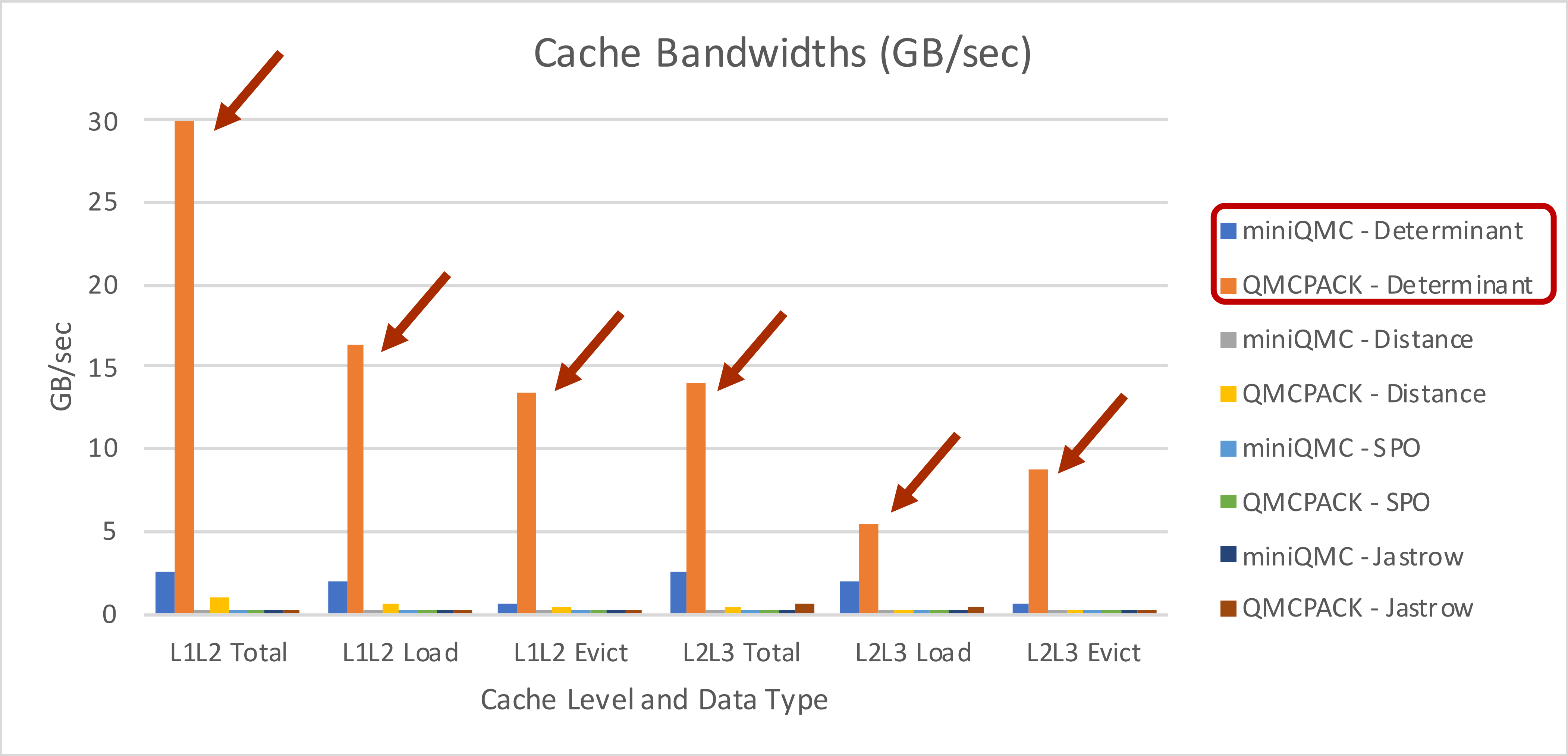
Have an inner loop that iterates over them

Branching Behavior Similarity

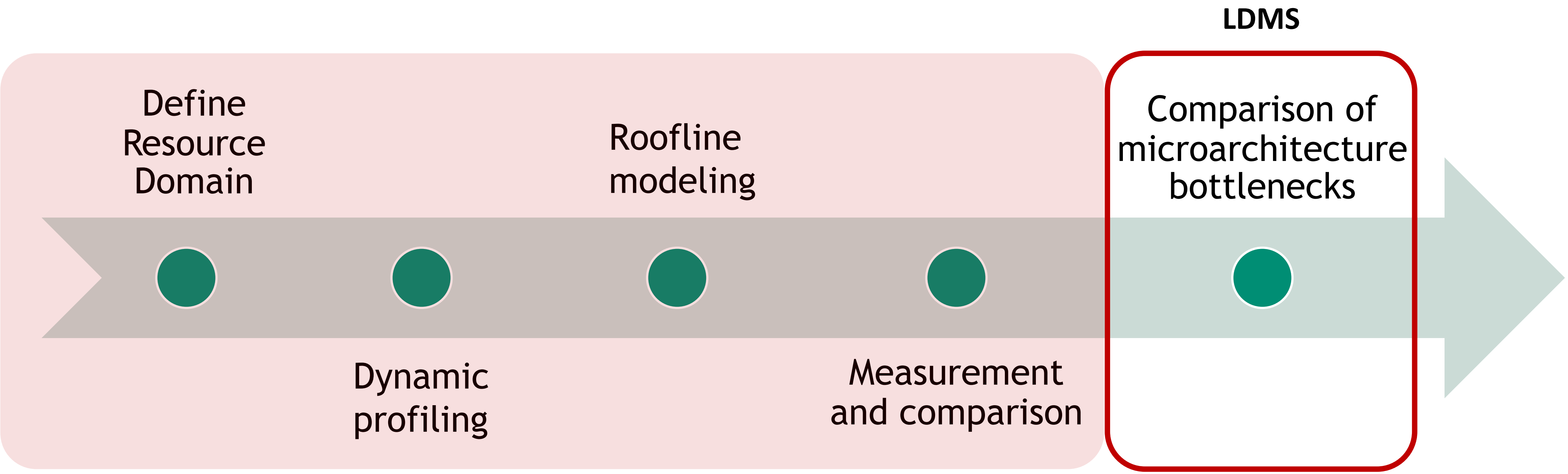


Jastrow kernel
&
SPO

Cache Behavior Similarity



Methodology



LDMS

Define Resource Domain

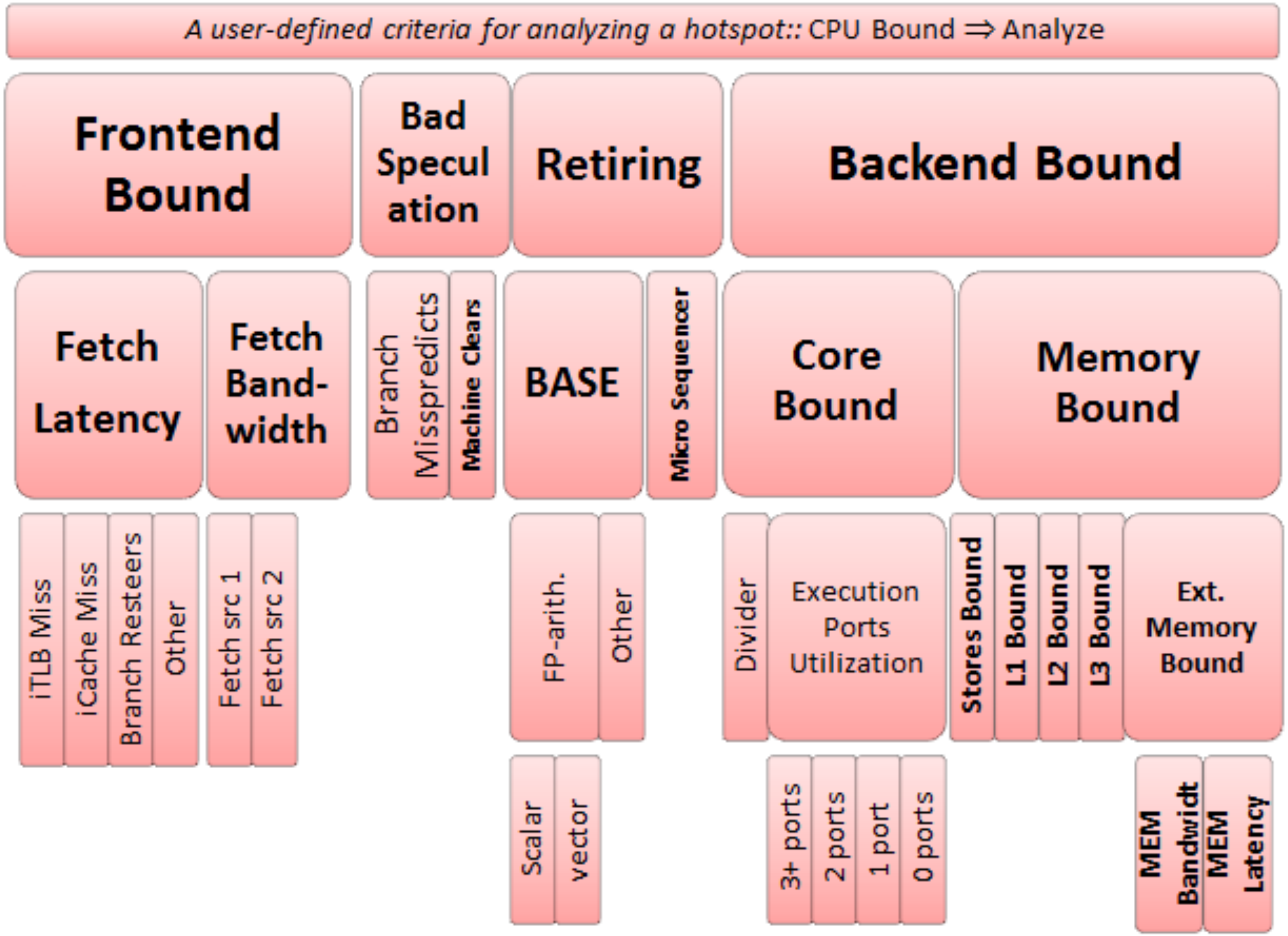
Dynamic profiling

Roofline modeling

Measurement and comparison

Comparison of microarchitecture bottlenecks

Hardware-Level Bottlenecks using TMA



A user-defined criteria for analyzing a hotspot:: CPU Bound ⇒ Analyze

Frontend Bound

Fetch Latency

iTLB Miss

iCache Miss

Branch Resteers

Other

Fetch Band-width

Fetch src 1

Fetch src 2

Bad Speculation

Branch Misspredicts

Machine Clears

Retiring

BASE

FP-arith.

Scalar

vector

Other

Micro Sequencer

Backend Bound

Core Bound

Divider

Execution Ports Utilization

3+ ports

2 ports

1 port

0 ports

Memory Bound

Stores Bound

L1 Bound

L2 Bound

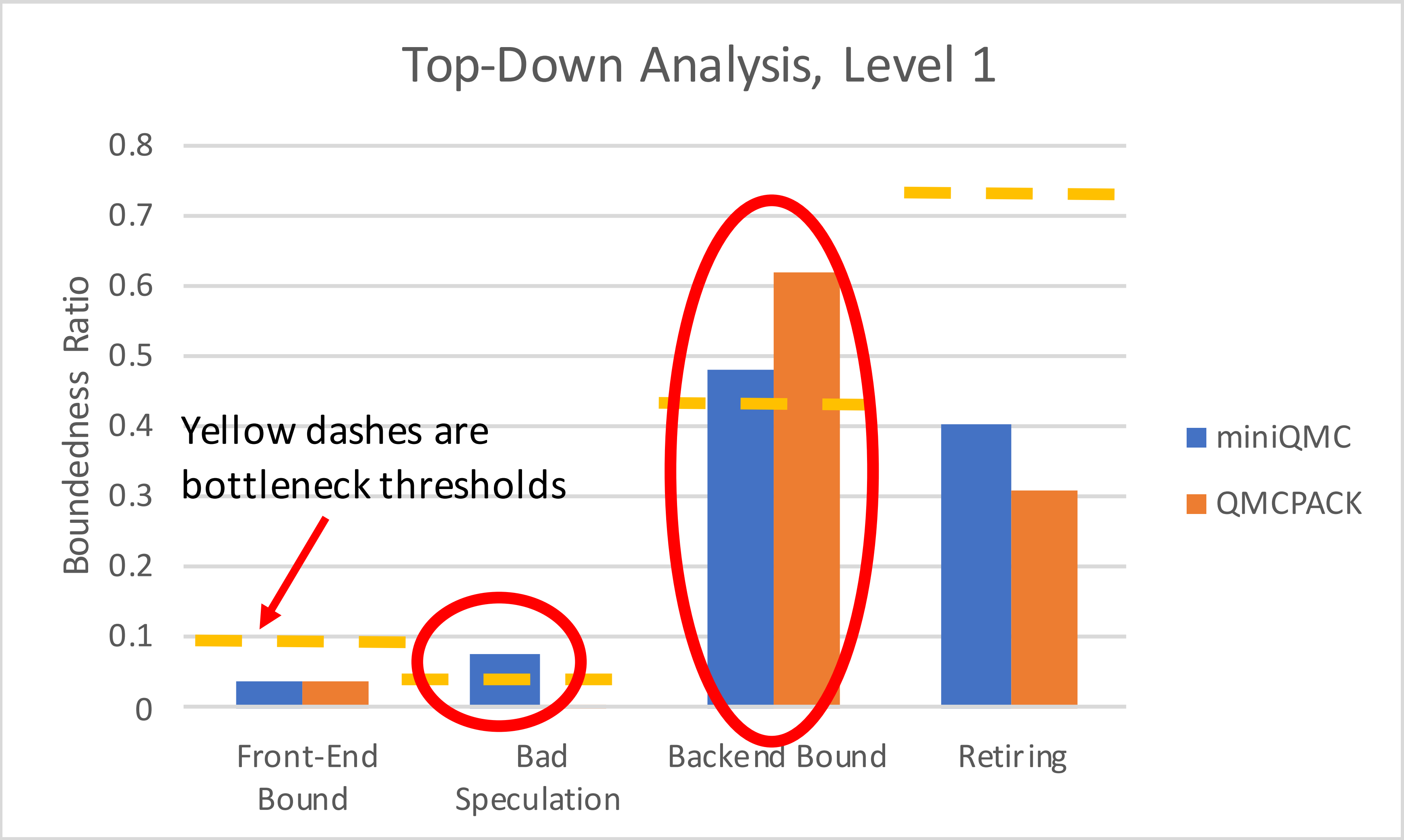
L3 Bound

Ext. Memory Bound

MEM Bandwidth

MEM Latency

TMA Level I



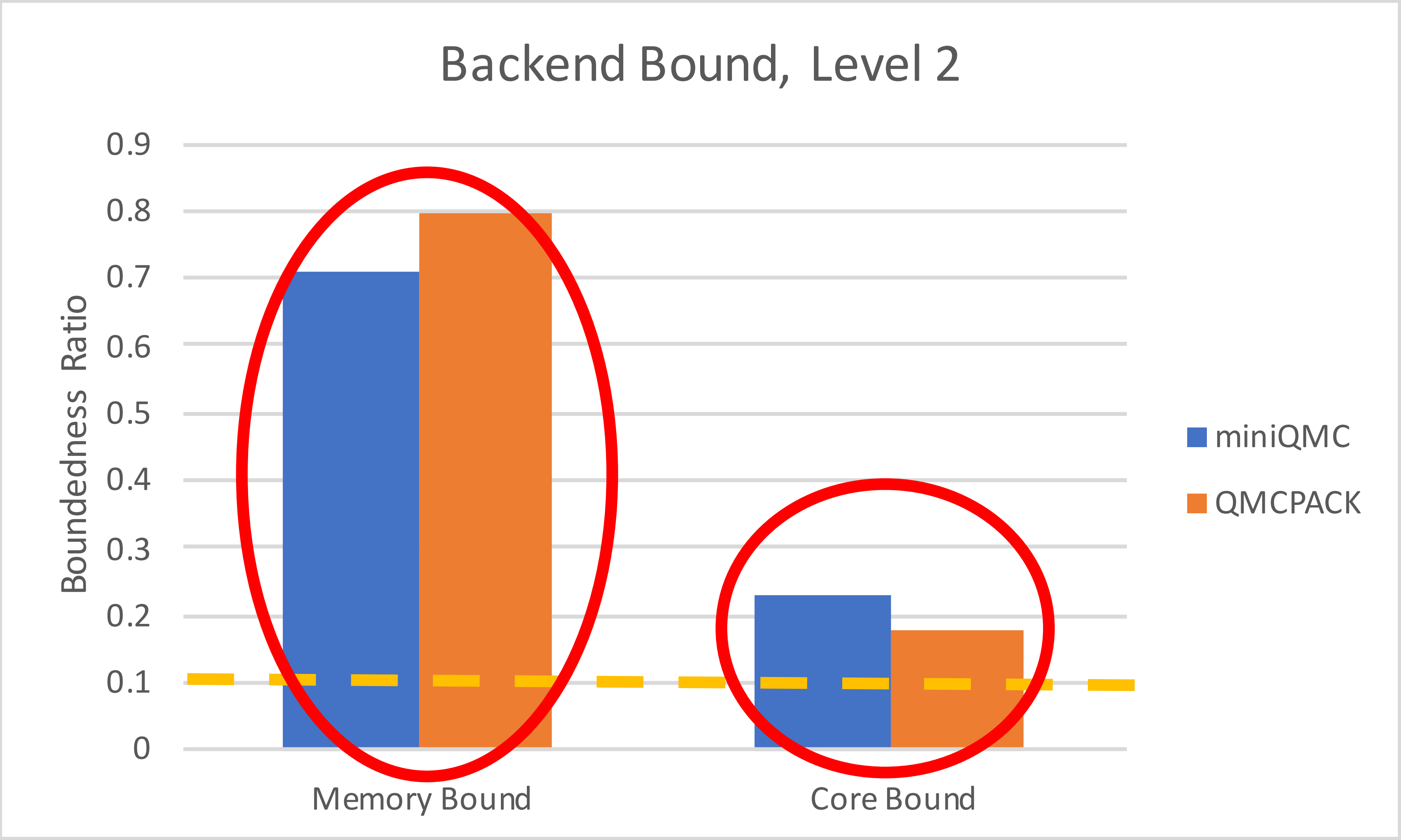
Yellow dashes are bottleneck thresholds

miniQMC & QMCPACK

Backend Bound

Due to micro-ops are not being delivered to the issue pipe

TMA Level 2 – Backend Bound

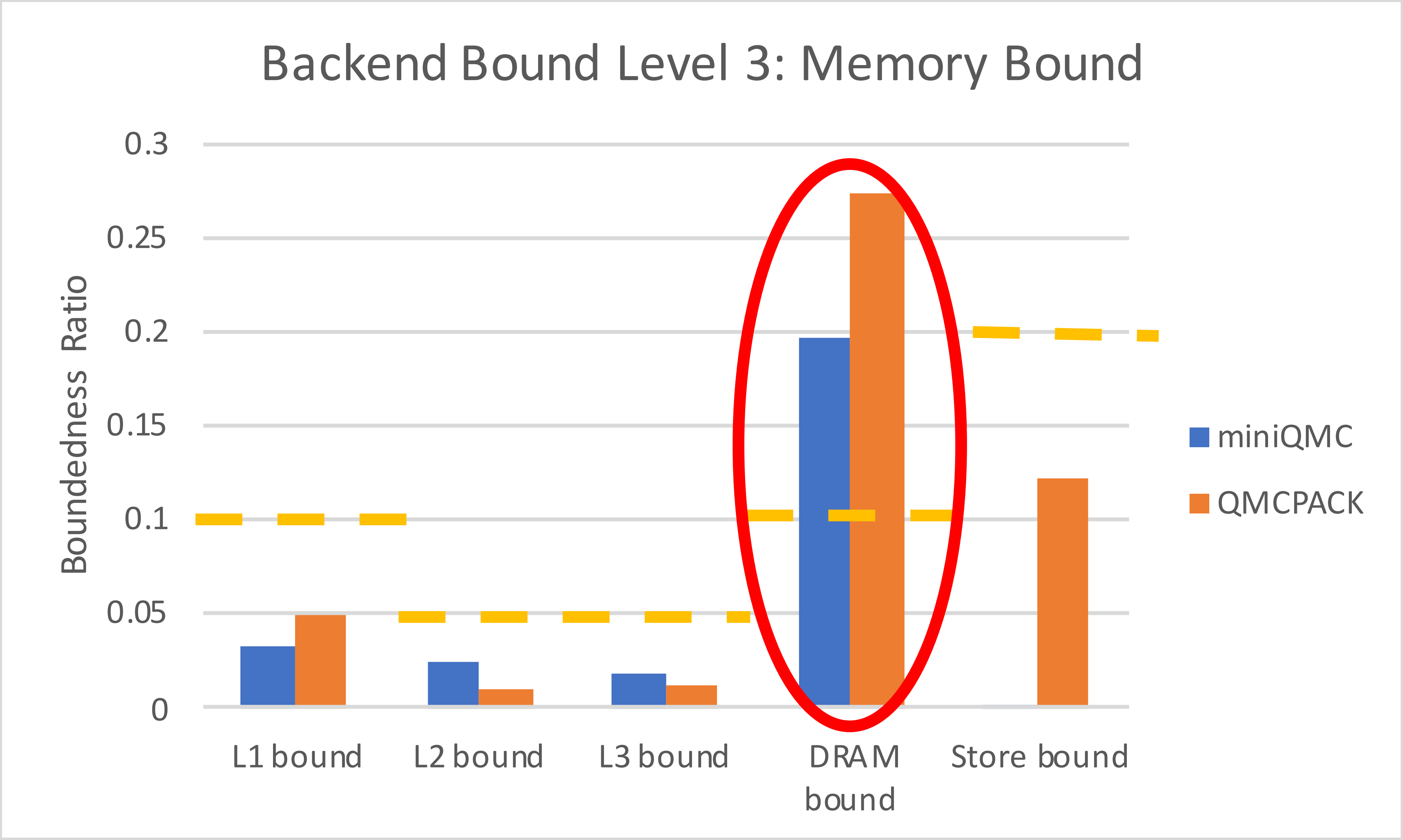


miniQMC & QMCPACK

Memory Bound

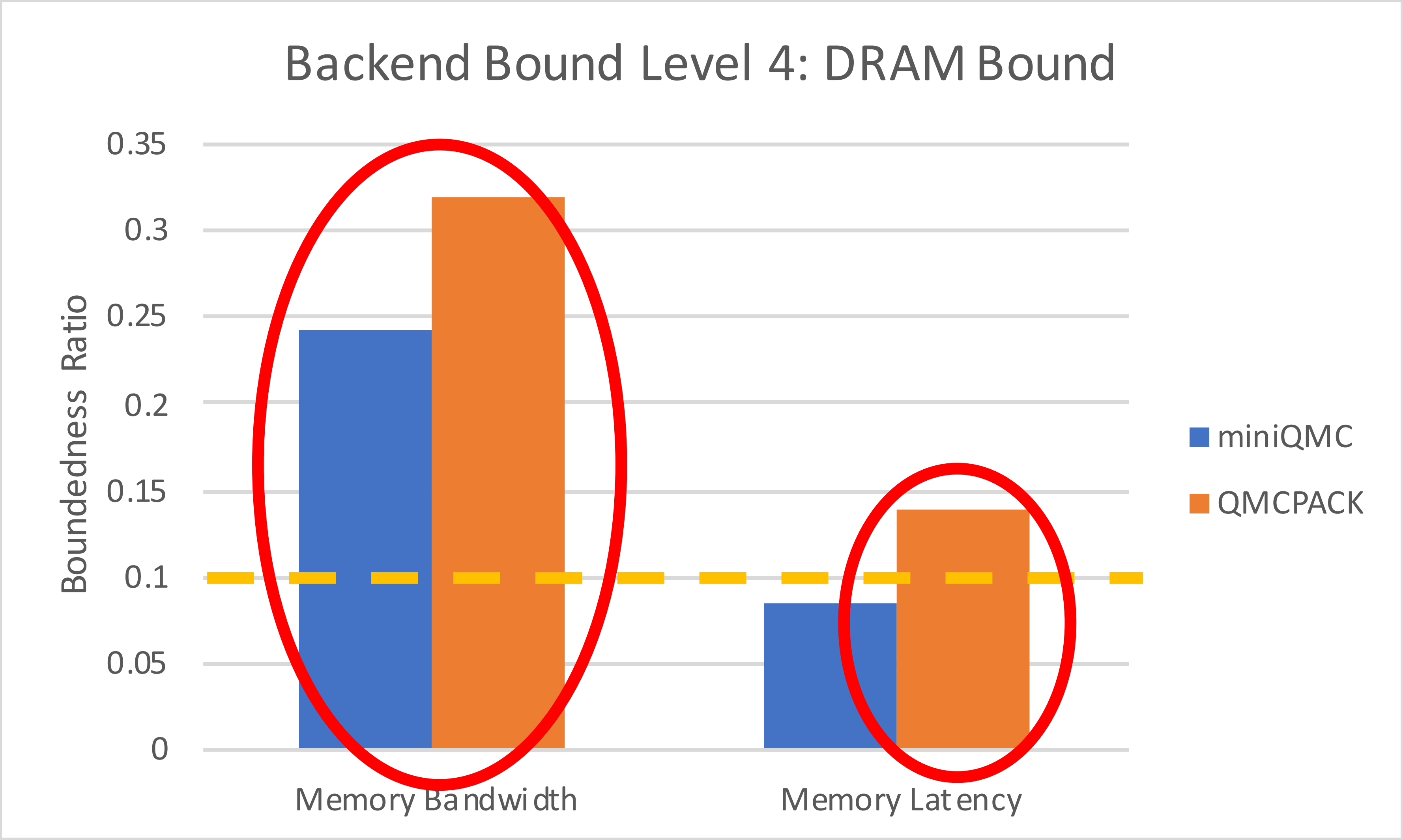
Slightly Core Bound

TMA Level 3 – Memory Bound



miniQMC & QMCPACK
DRAM Bound

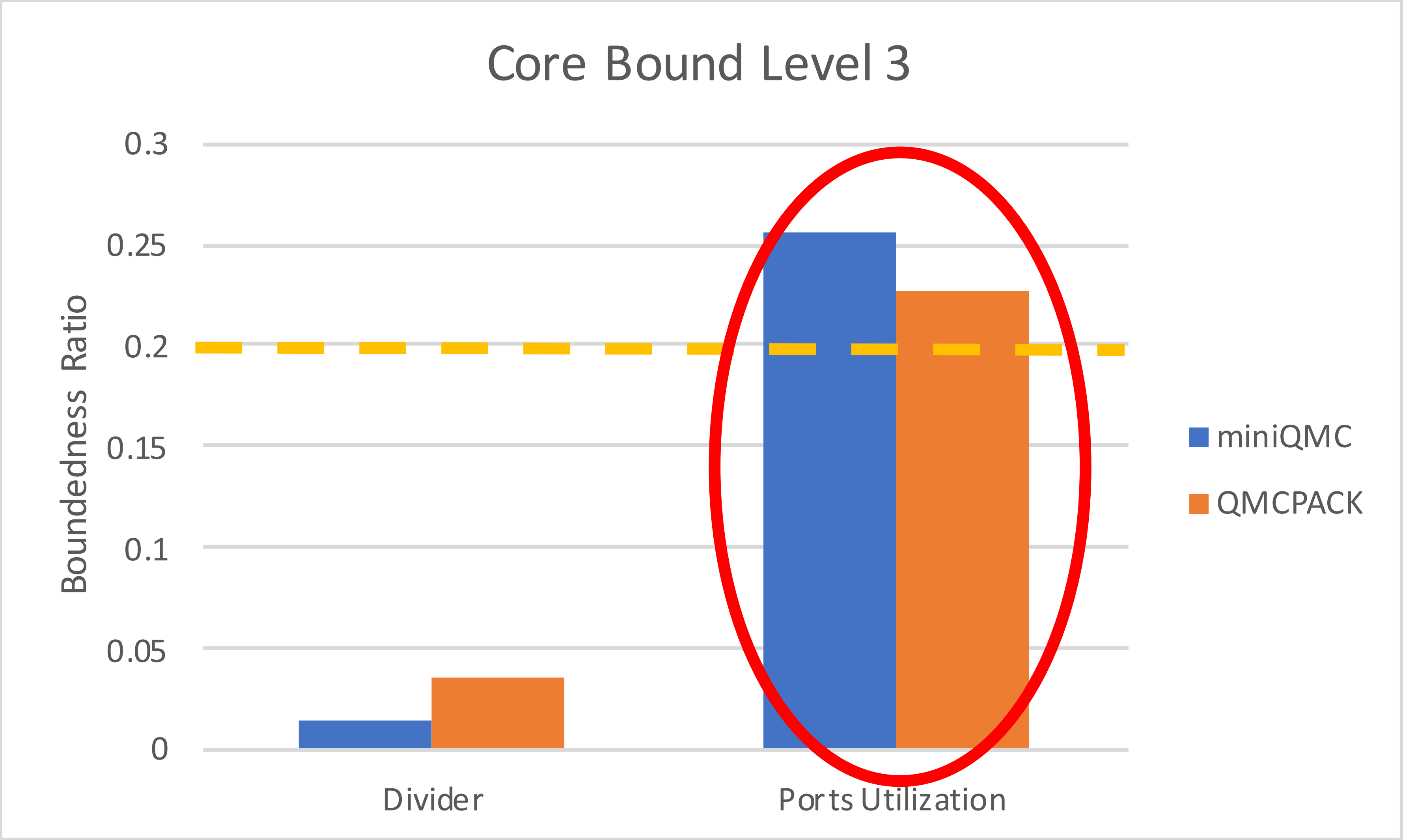
TMA Level 4 – DRAM Bound



miniQMC & QMCPACK

Bandwidth!

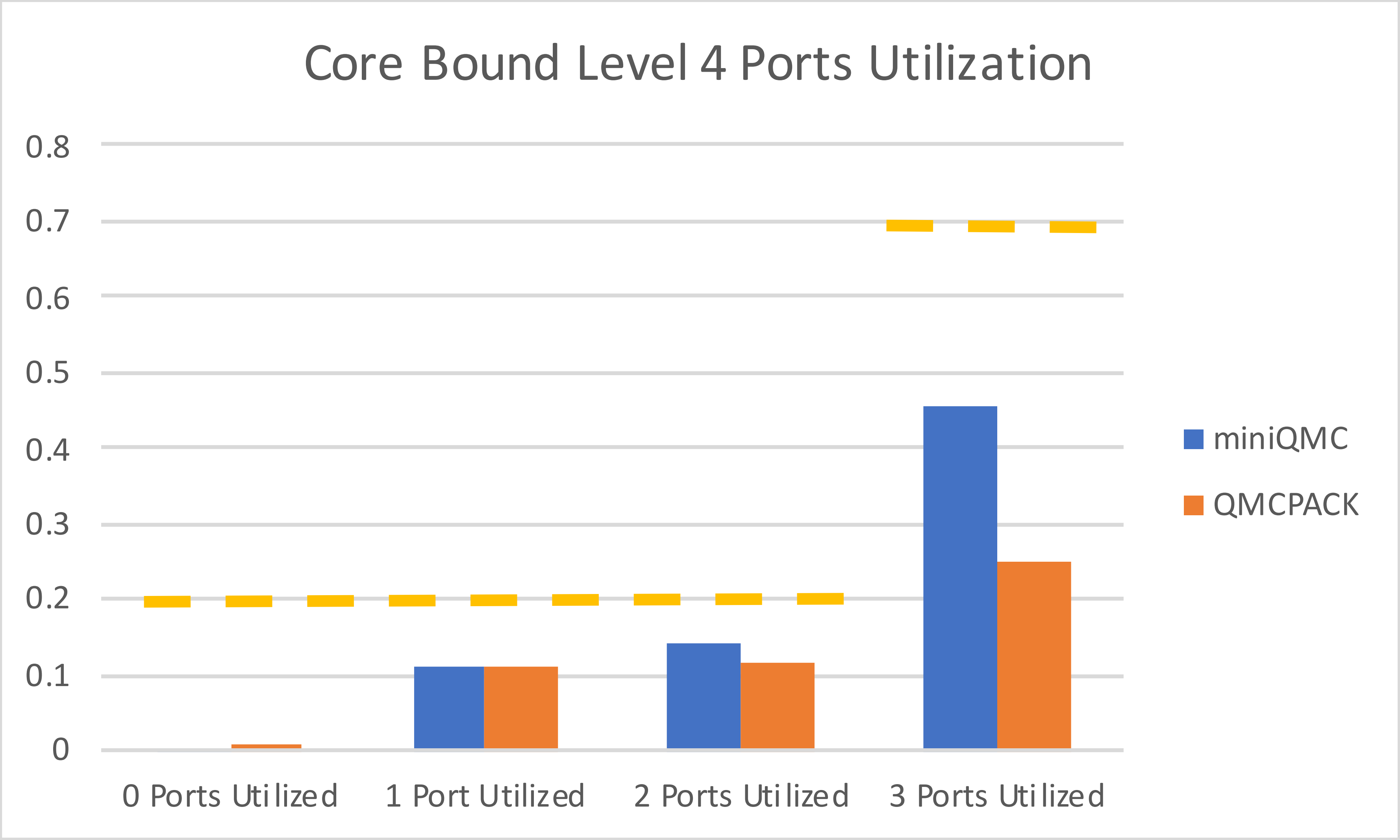
TMA Level 3 – Core Bound



miniQMC & QMCPACK

Port Utilization

TMA Level 4 – Port Utilization



miniQMC & QMCPACK

TMA couldn’t capture the problem ☹

Is miniQMC a Good Proxy for QMCPACK?

Depends on how the proxy will be used

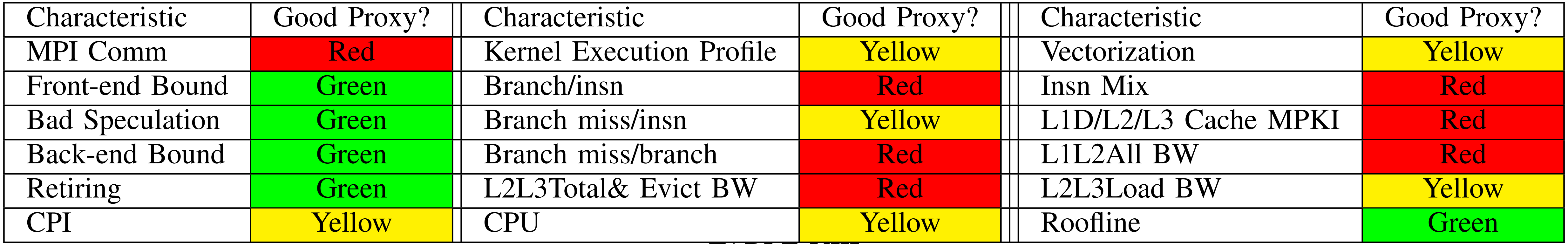
miniQMC is a good proxy for QMCPACK for certain cases

Does not faithfully model QMCPACK in every aspect

Good in term of:

- Whole-application level
- Hardware bottlenecks
- Kernel-only execution profiles

Is miniQMC a Good Proxy for QMCPACK?



Characteristic	Good Proxy?	Characteristic	Good Proxy?	Characteristic	Good Proxy?
MPI Comm	Red	Kernel Execution Profile	Yellow	Vectorization	Yellow
Front-end Bound	Green	Branch/insn	Red	Insn Mix	Red
Bad Speculation	Green	Branch miss/insn	Yellow	L1D/L2/L3 Cache MPKI	Red
Back-end Bound	Green	Branch miss/branch	Red	L1L2All BW	Red
Retiring	Green	L2L3Total& Evict BW	Red	L2L3Load BW	Yellow
CPI	Yellow	CPU	Yellow	Roofline	Green

🟩 REPRESENTATIVE

🟨 PARTIALLY REPRESENTATIVE

🟥 NOT REPRESENTATIVE

Future Work

Examine alternative methods

Incorporate the work on the communication patterns by exploring pairs with communication implementation

Explore more parent/proxy pairs

Acknowledgment

This research was supported by the **Exascale Computing Project (ECP)**, Project Number 17-SC-20-SC, a collaborative effort of two DOE organizations, the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem including software, applications, hardware, advanced system engineering, and early testbed platforms, to support the nation's exascale computing imperative.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE- NA0003525.

Questions ?

What's new?

- Dynamic profiling
- Roofline modeling
- Qualitative comparison of quantitative metrics
- Intel’s Top-Down Microarchitecture Analysis (TMA)