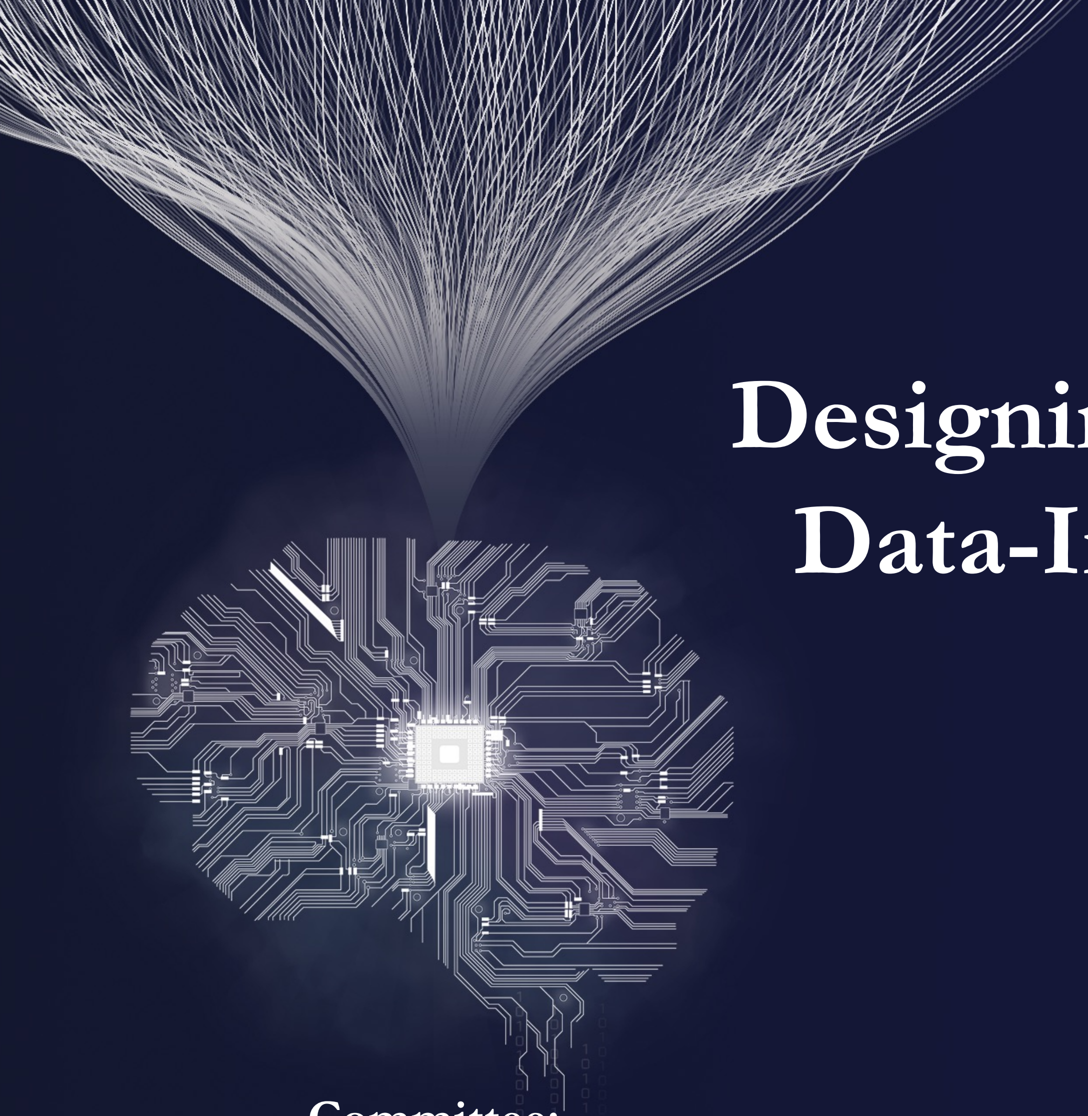


SAFARI

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Designing, Modeling, and Optimizing Data-Intensive Computing Systems

Gagandeep Singh

Ph.D. Defense

Committee:

Henk Corporaal (TU Eindhoven)
Onur Mutlu (ETH, Zurich)
Sander Stuijk (TU Eindhoven)
C.H. Berkel (TU Eindhoven)
Peter Hofstee (IBM Austin/TU Delft)
Francky Catthoor (IMEC/KU Leuven)
Dionysios Diamantapoulos (IBM Research Europe)
Osman Unsal (BSC)

COSMO
10ZB

SKA
300PB

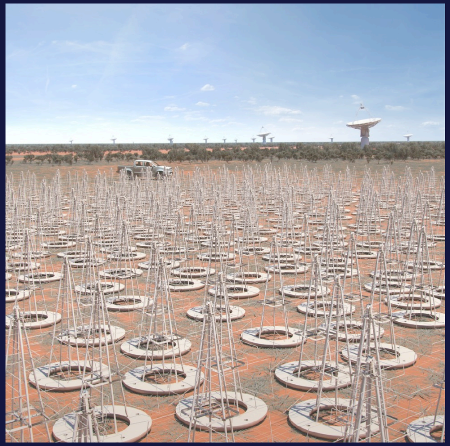
uploads on
facebook
180PB

searches on
Google
98PB

YouTube
15PB

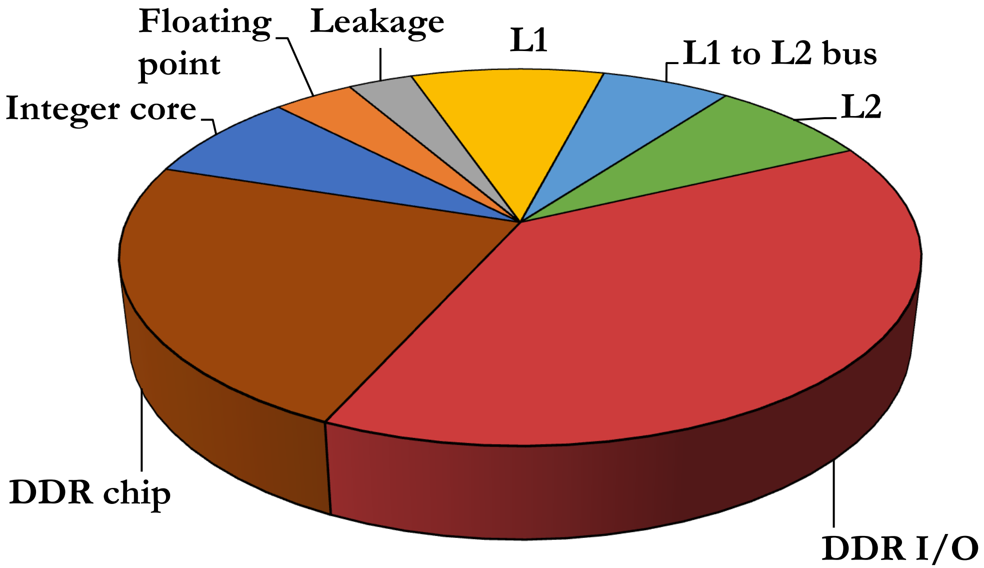
CERN
15PB

NASDAQ
3PB



Michael Wise, ASTRON, "Science data centre challenges", *DOME Symposium*, May, 2017
NVIDIA/MeteoSwiss: An example of COSMO simulation with cloud patterns over Switzerland and surrounding areas
C. Schär et al., "Kilometer-scale climate models: Prospects and challenges," *BAMS*, 2020

Current Computing Systems

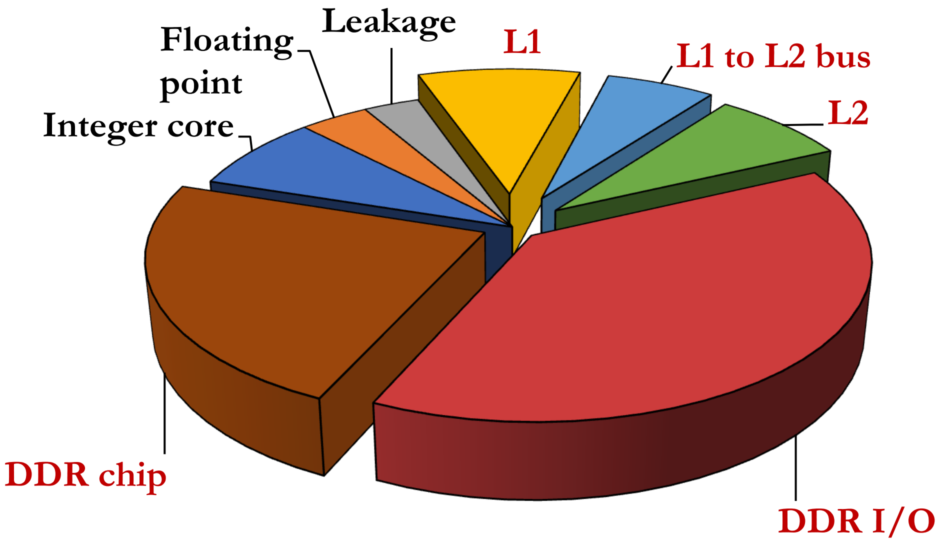


System-level energy
break down

R. Nair et al., "Active memory cube: A processing-in memory architecture for exascale systems", *2nd Workshop on Near-Data Processing*, 2014
A. Boroumand et al., "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks", *ASPLOS*, 2018

Current Computing Systems

- **Data movement** dominates **energy consumption**
  - Especially **off-chip data movement**



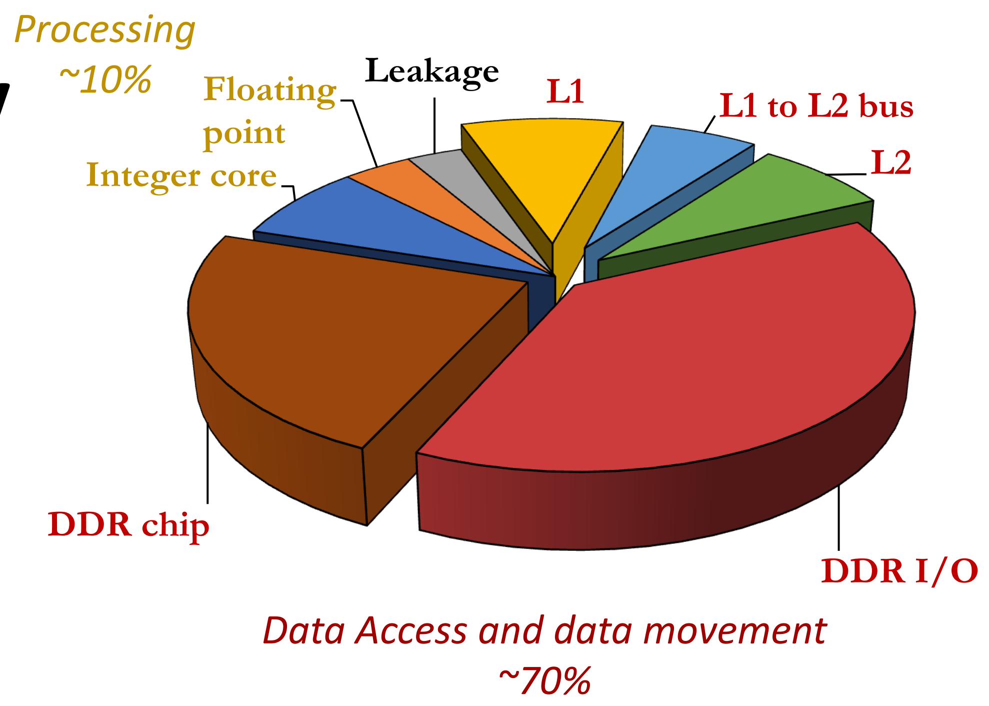
Data Access and data movement ~70%

System-level energy break down

R. Nair et al., "Active memory cube: A processing-in memory architecture for exascale systems", *2nd Workshop on Near-Data Processing*, 2014
A. Boroumand et al., "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks", *ASPLOS*, 2018

Current Computing Systems

- **Data movement** dominates **energy consumption**
  - Especially **off-chip data movement**

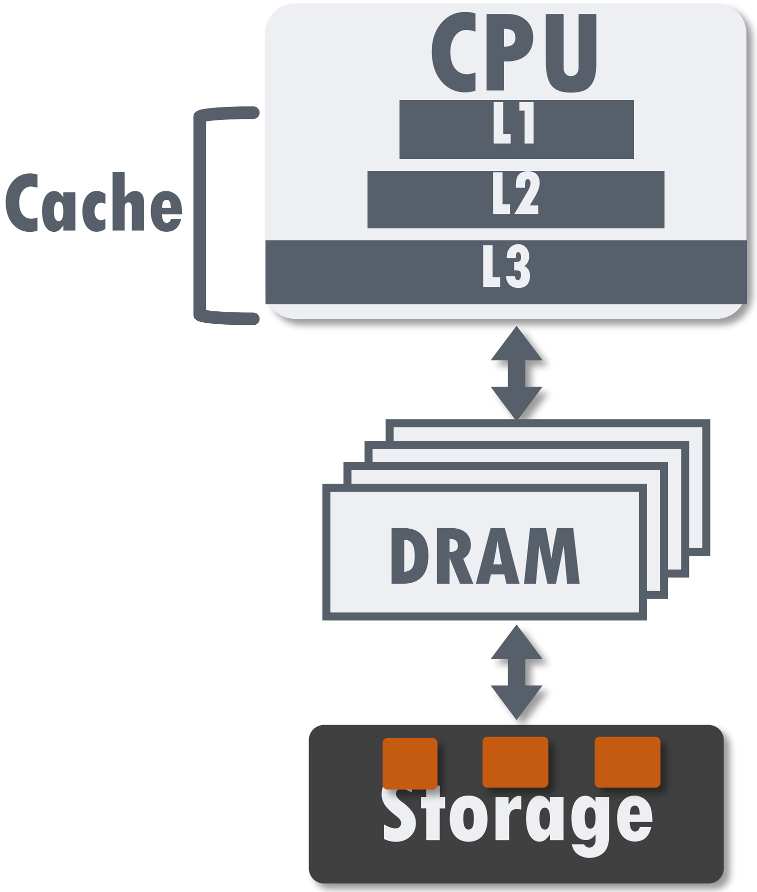


System-level energy break down

R. Nair et al., "Active memory cube: A processing-in memory architecture for exascale systems", *2nd Workshop on Near-Data Processing*, 2014
A. Boroumand et al., "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks", *ASPLOS*, 2018

Current Computing Systems

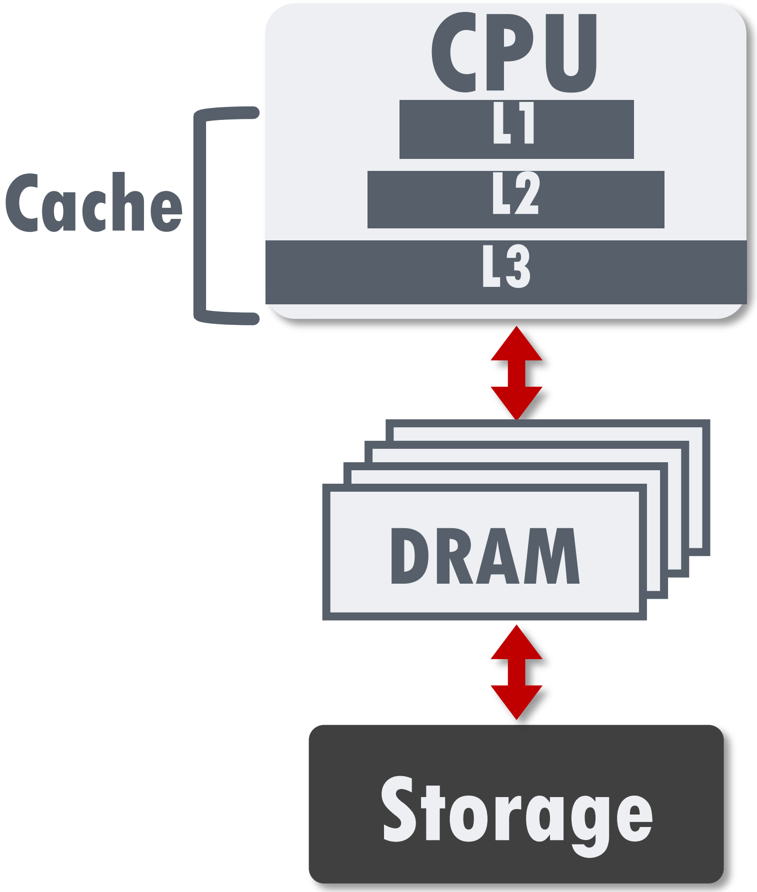
- **Data movement** dominates **energy consumption**
  - Especially **off-chip data movement**



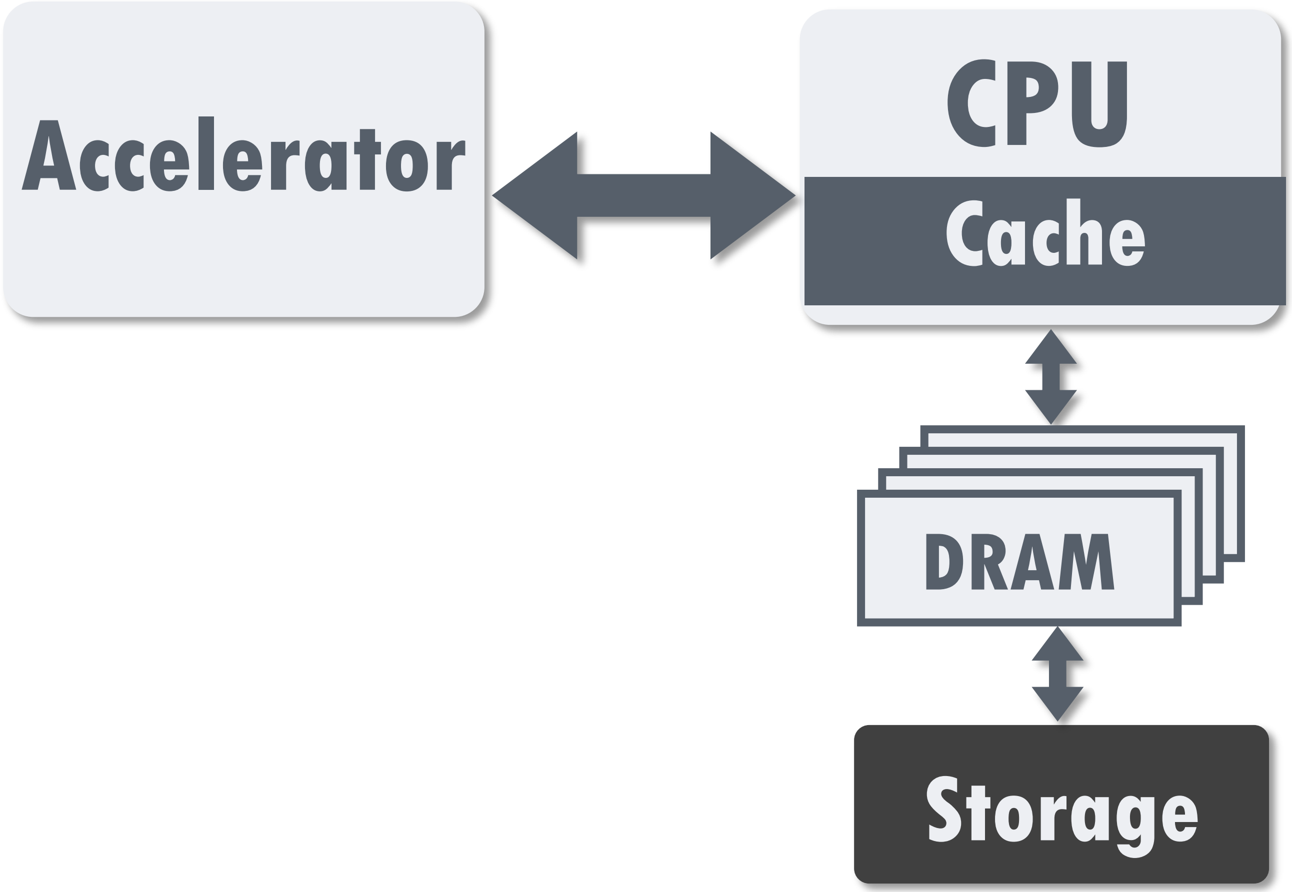
Current Computing Systems

- **Data movement** dominates **energy consumption**
    - Especially **off-chip data movement**

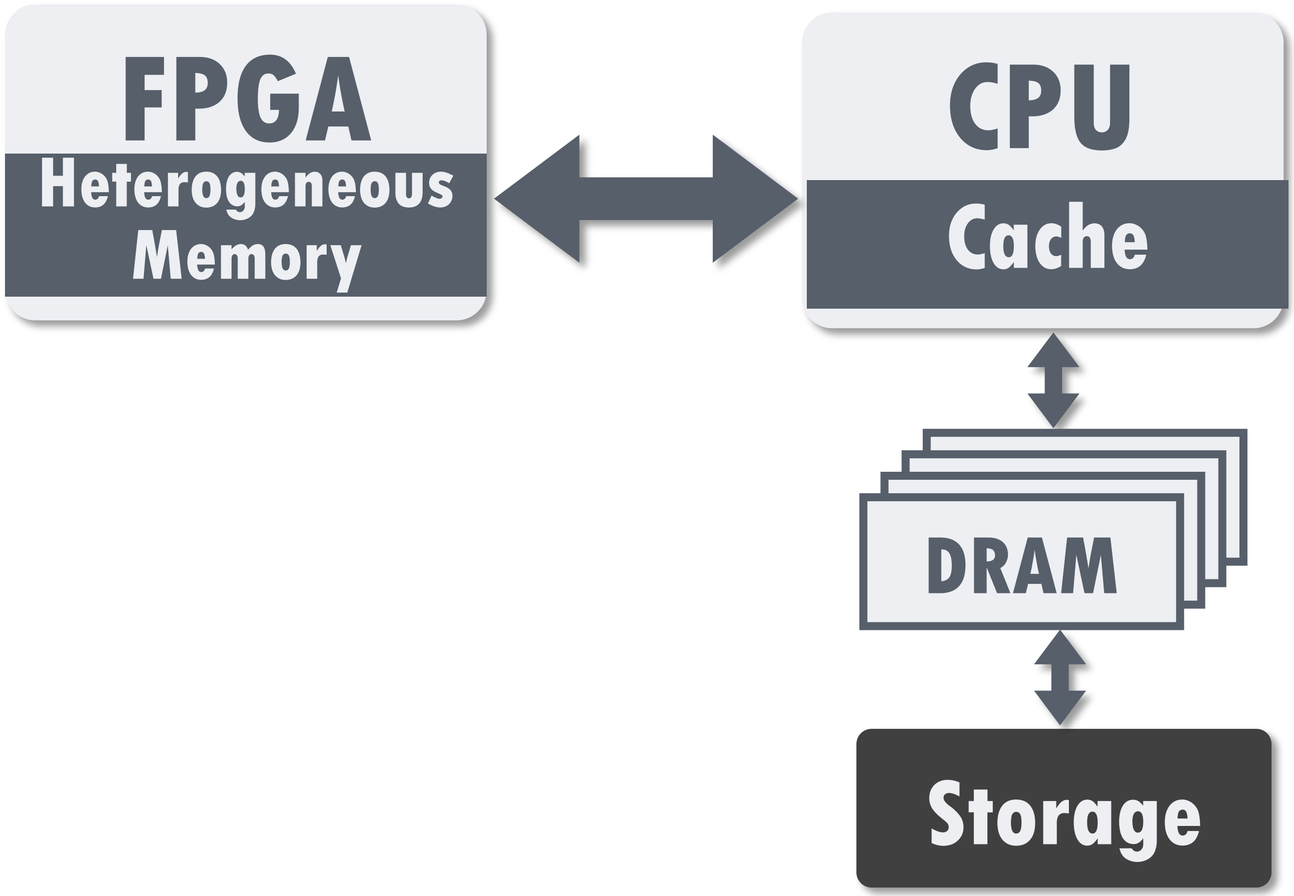
- **Data-intensive workloads** are **memory-bound**



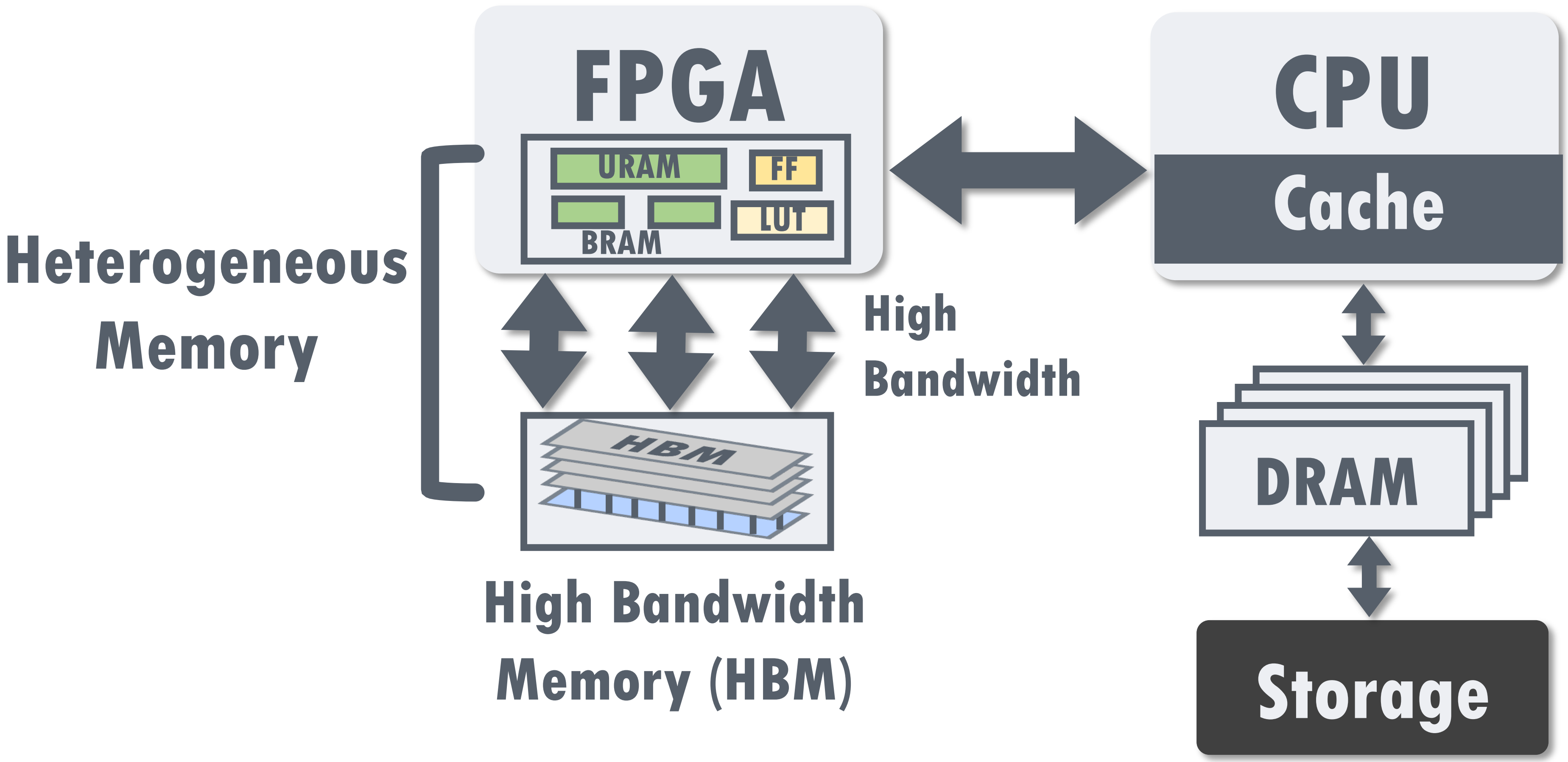
Data-Centric Computing



Data-Centric Computing



Data-Centric Computing



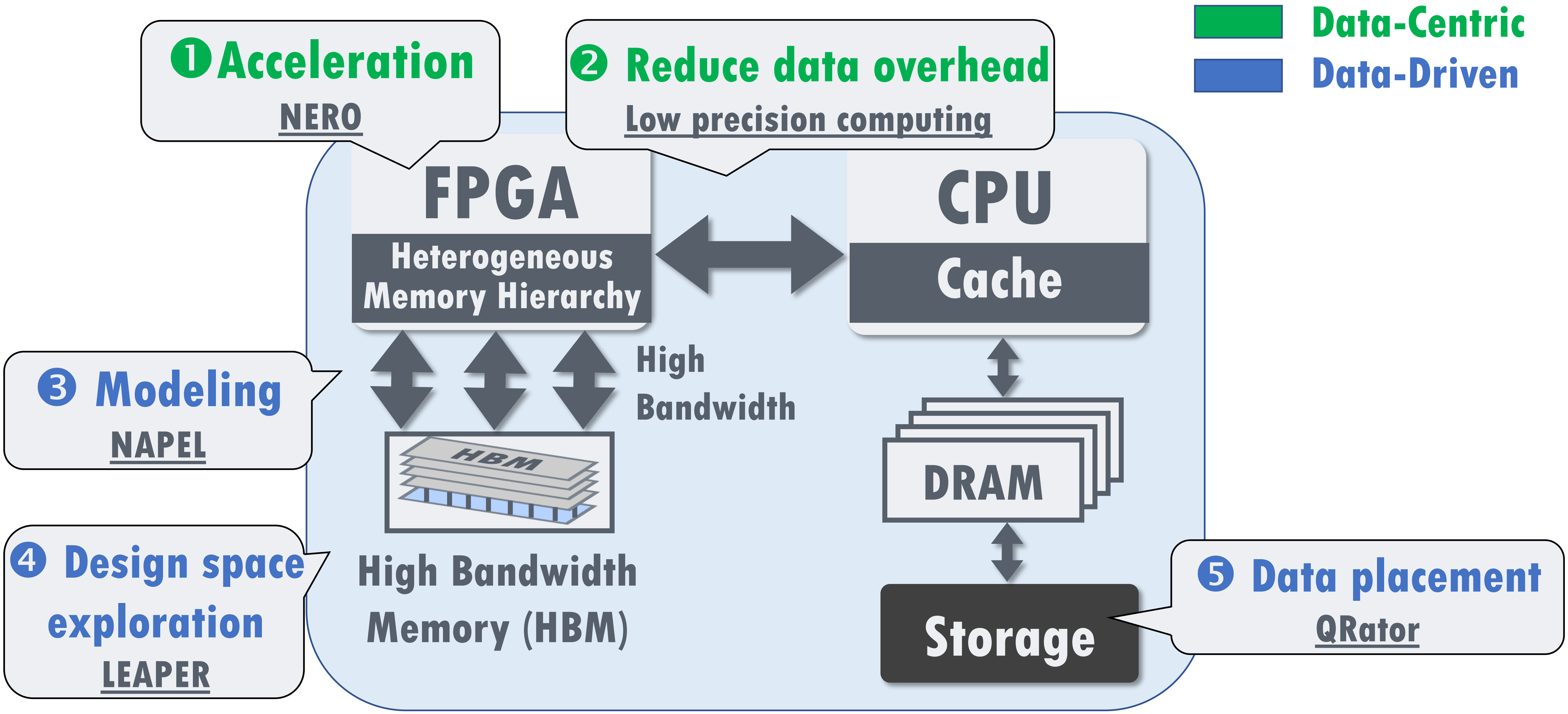
Thesis Statement

Design system architectures to **effectively handle data** by:

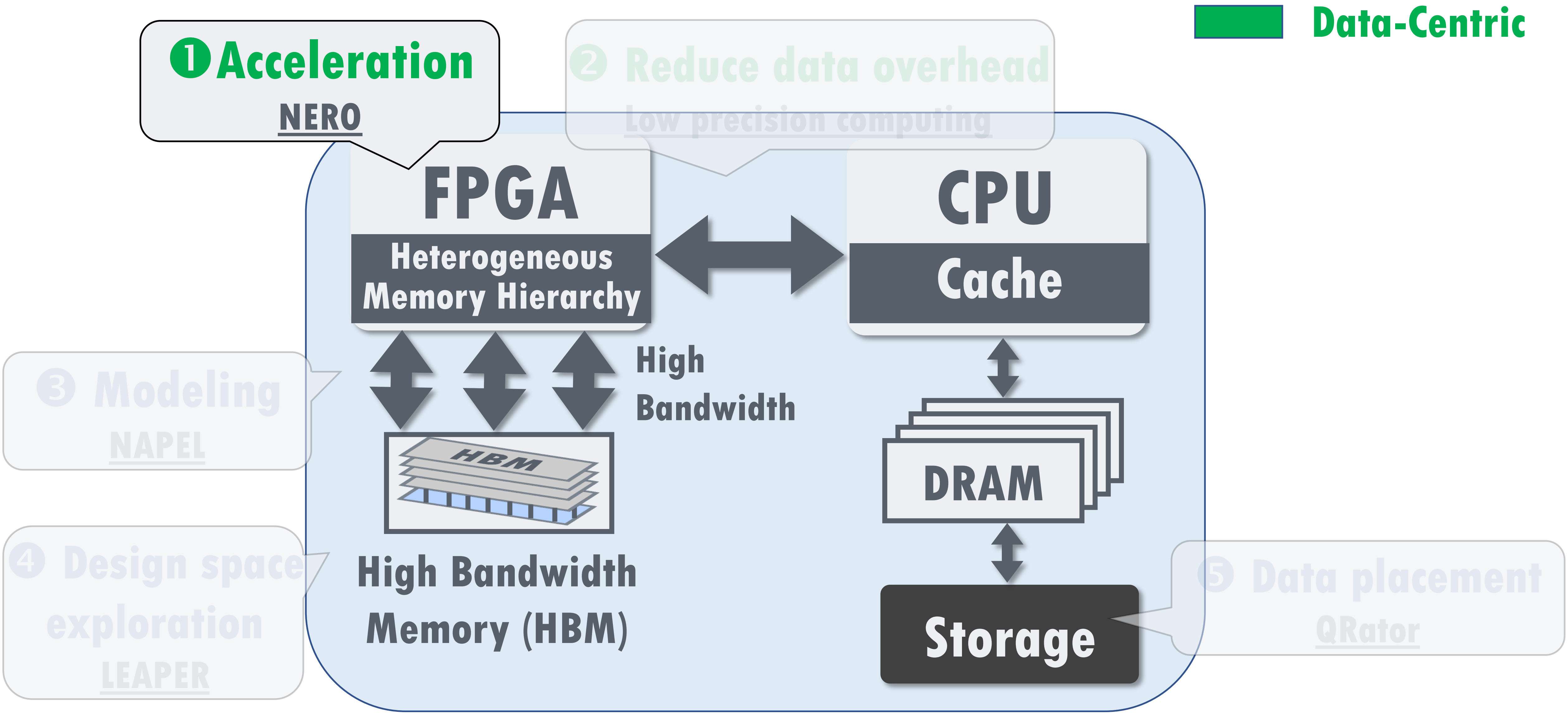
Data-centric approach

Data-driven approach

Thesis Contributions



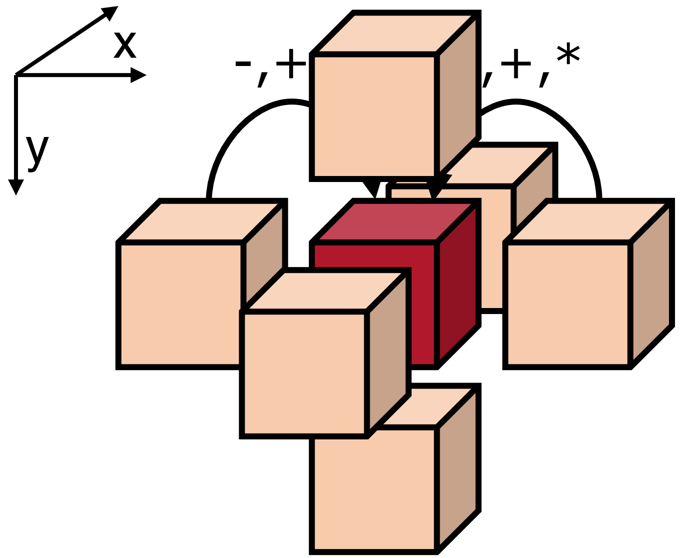
(1) NERO: Weather Prediction Accelerator



(1) NERO: Weather Prediction Accelerator

Key: Stencil computation

- **Complex memory-access patterns**



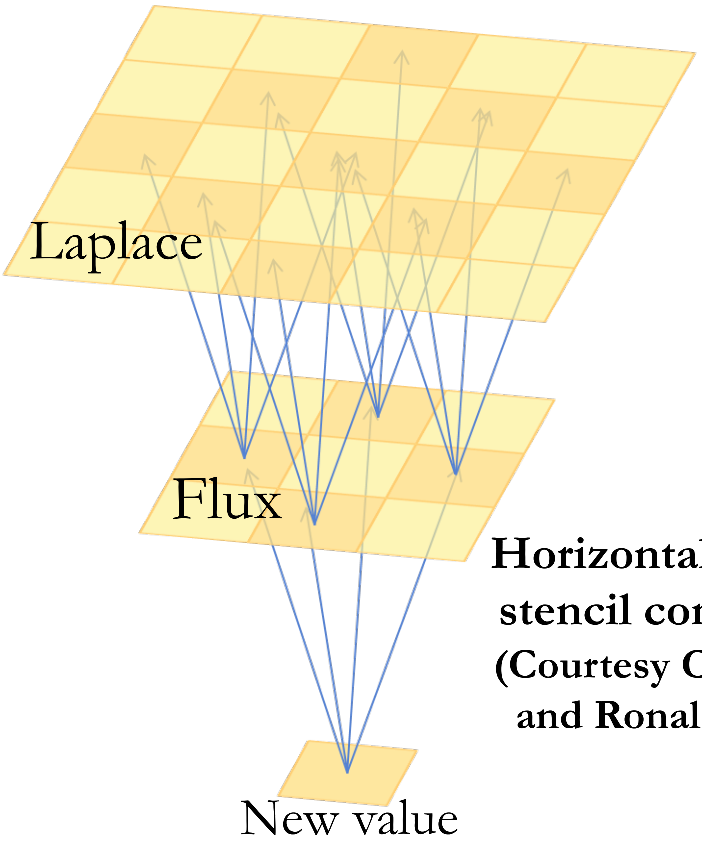
e.g., 7-point Jacobi in 3D plane

(1) NERO: Weather Prediction Accelerator

Key: Stencil computation

- **Complex memory-access patterns**

- **~80 compound stencils**



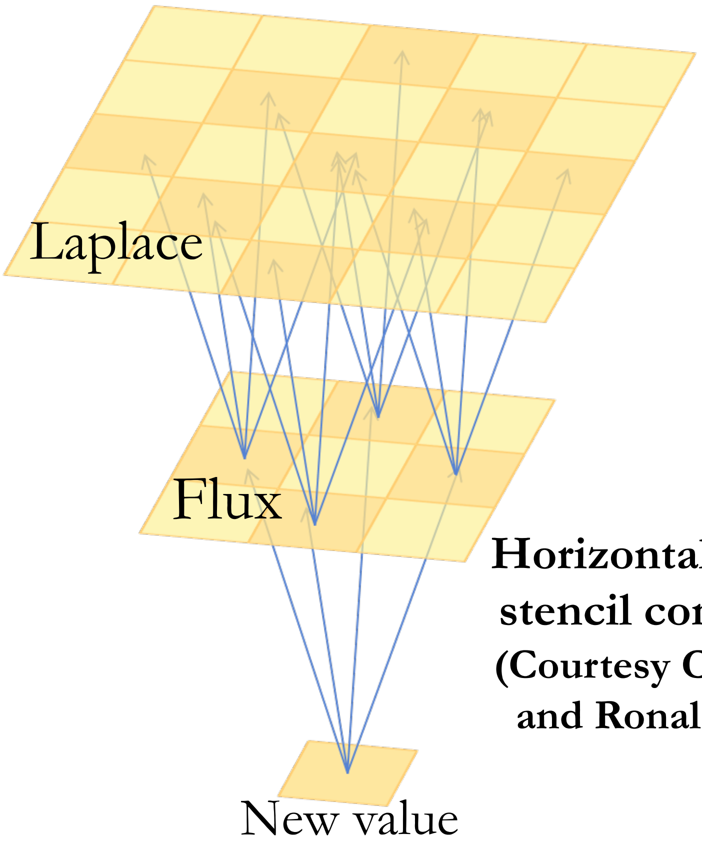
Horizontal diffusion stencil composition (Courtesy CSCS/ETH and Ronald Luijten)

(1) NERO: Weather Prediction Accelerator

Key: Stencil computation

- Complex memory-access patterns

- ~80 compound stencils



Horizontal diffusion stencil composition (Courtesy CSCS/ETH and Ronald Luijten)

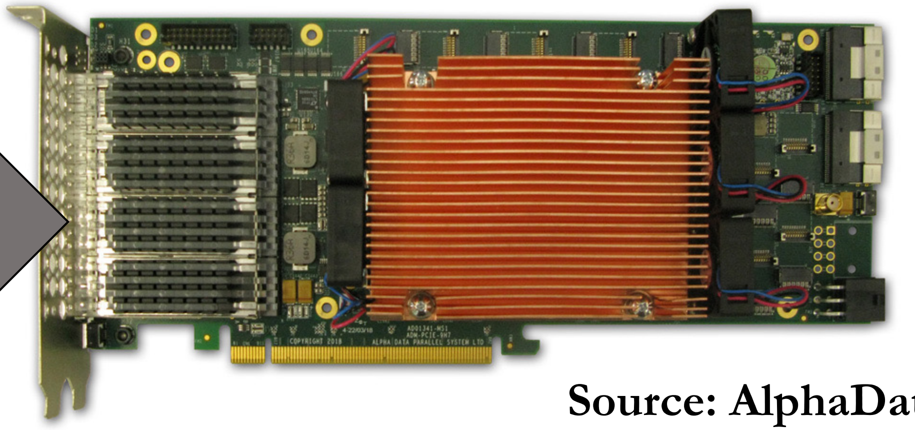
Memory bound with limited performance

(1) NERO: Weather Prediction Accelerator

Source: IBM

IBM POWER9 CPU

CAPI



Source: AlphaData

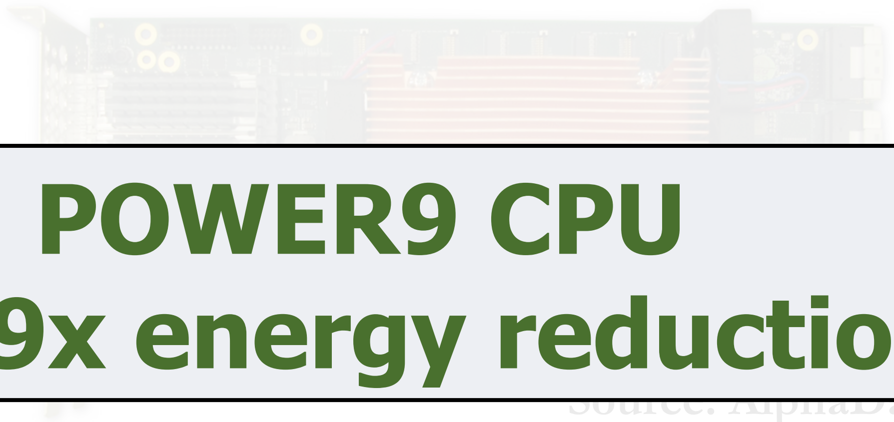
HBM-based FPGA board

Near-HBM FPGA-based accelerator

(1) NERO: Weather Prediction Accelerator

Source: IBM

IBM POWER9

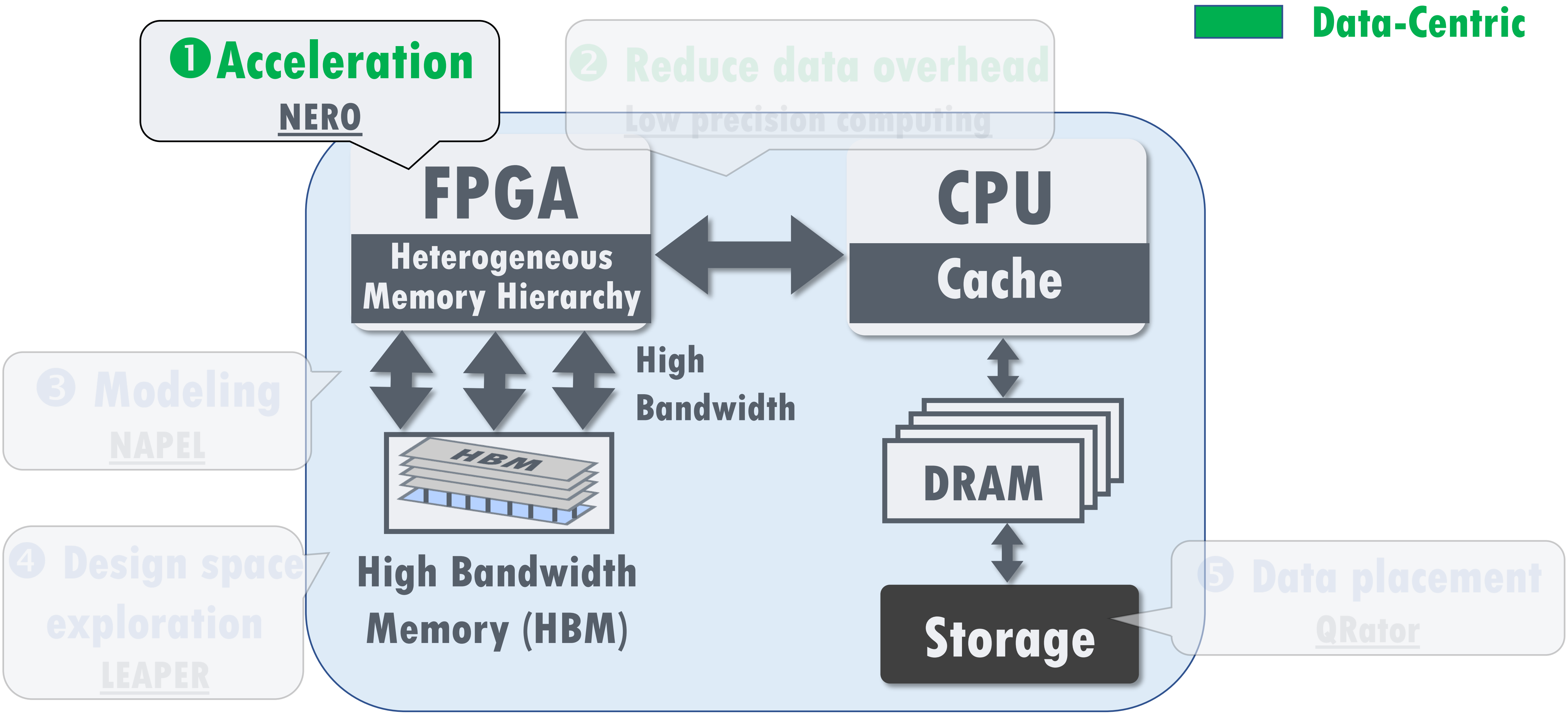


HBM-based FPGA board

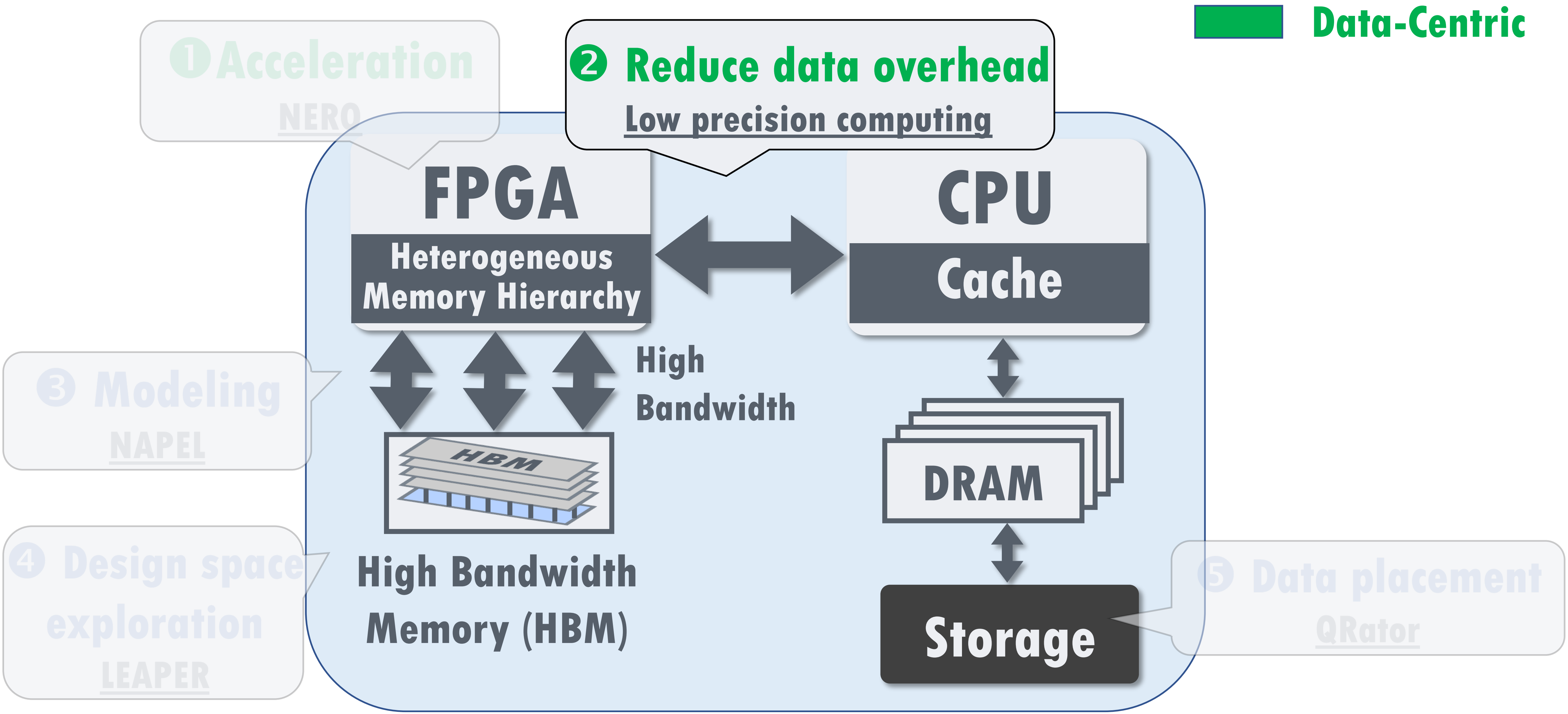
Compared to IBM POWER9 CPU
4x-8x faster with 22x-29x energy reductions

Energy efficiency of 1.5-17.3 GFLOPS/Watt

(1) NERO: Weather Prediction Accelerator



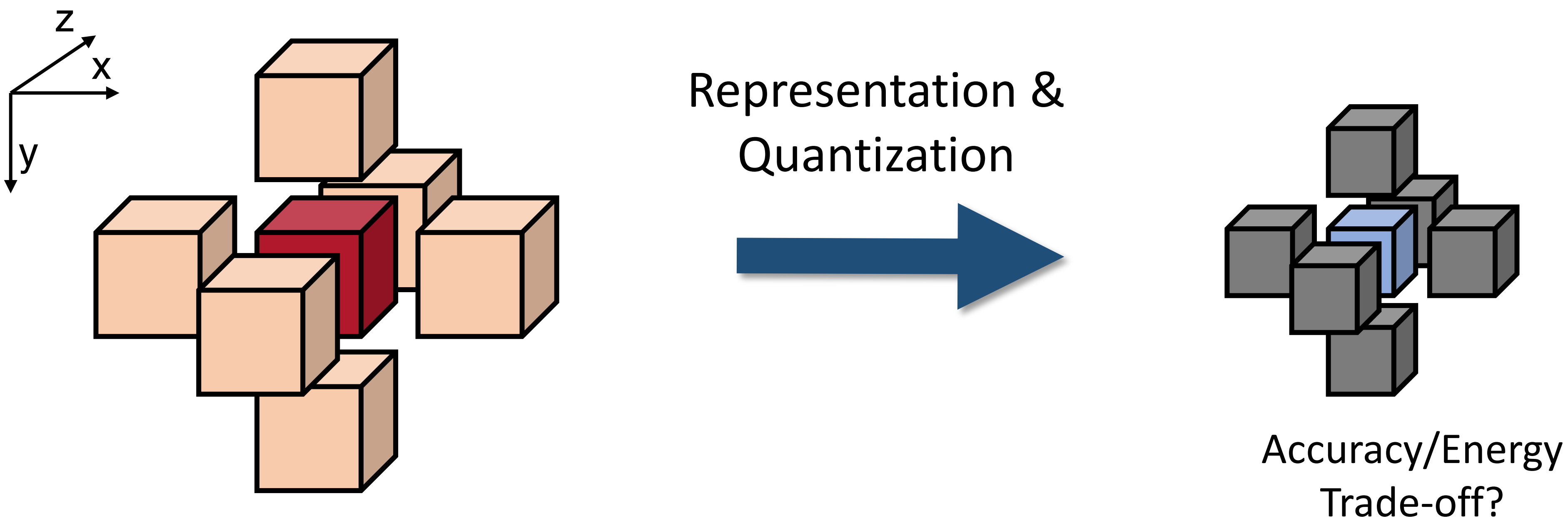
(2) Reduced-Precision Stencil Computation



(2) Reduced-Precision Stencil Computation

High-precision computation is costly

Requiring higher power, energy, and bandwidth



(2) Reduced-Precision Stencil Computation

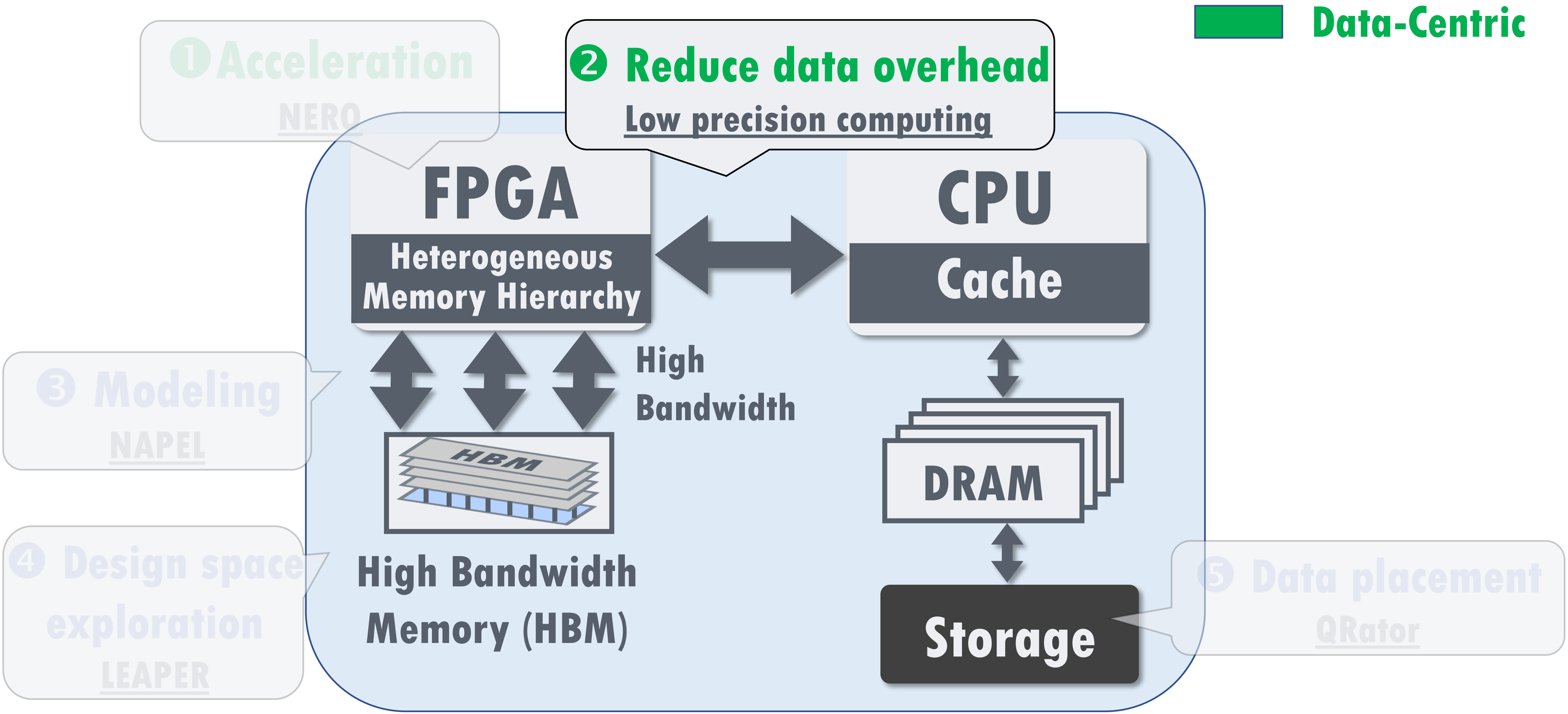
High-precision number format are costly :

50% fewer bits with only 1% loss of accuracy

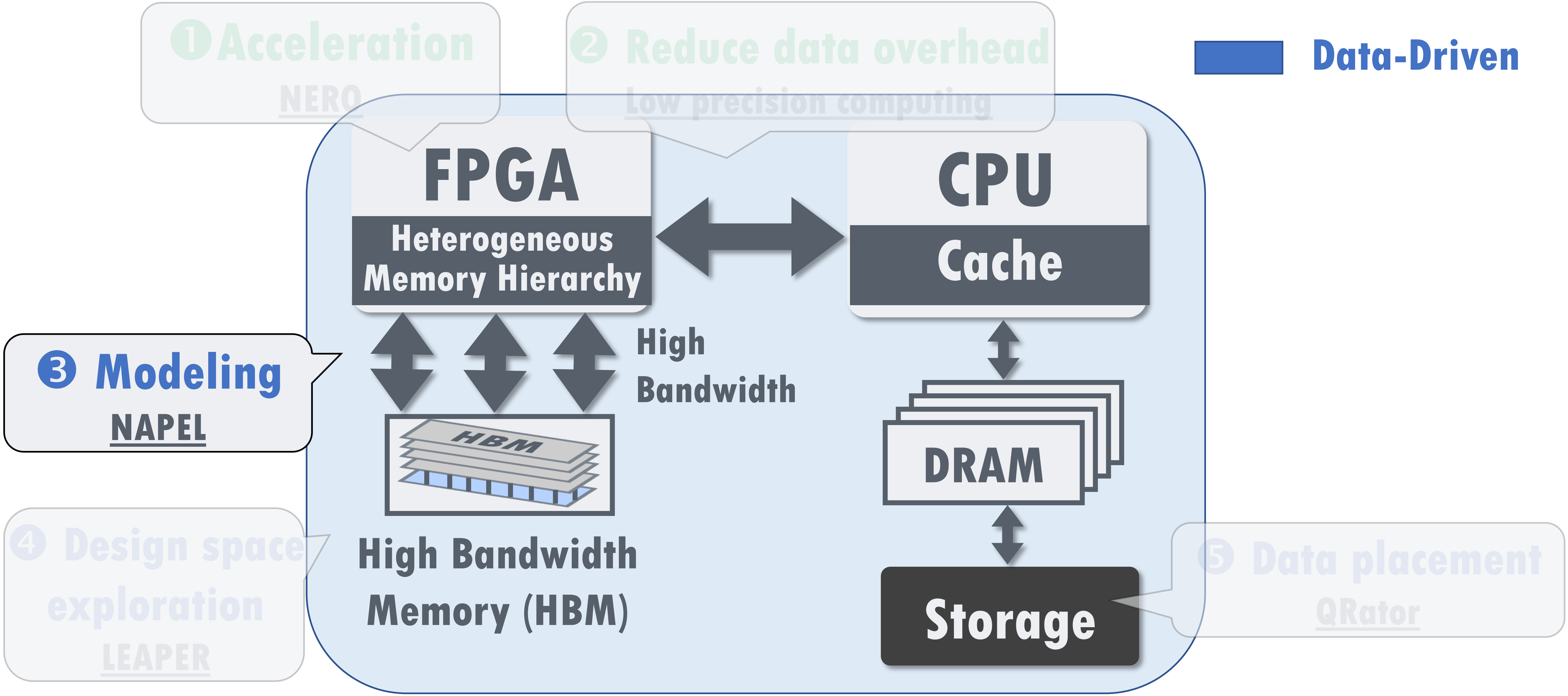
30-50x higher energy efficiency

Accuracy/Energy trade-off?

(2) Reduced-Precision Stencil Computation



(3) NAPEL: ML-Based Simulation



(3) NAPEL: ML-Based Simulation

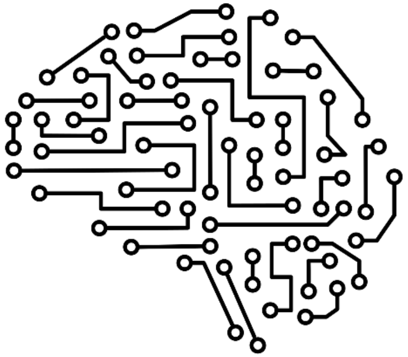
Early-stage simulation:

- **Workload suitability analysis**
- **Design space exploration (DSE)**
- Example Simulators: Sinuca[2015], Gem5+HMC[2017], Ramulator-PIM[2019]

Simulation of real workloads can be 10000x slower than native-execution!!!

(3) NAPEL: ML-Based Simulation

Application

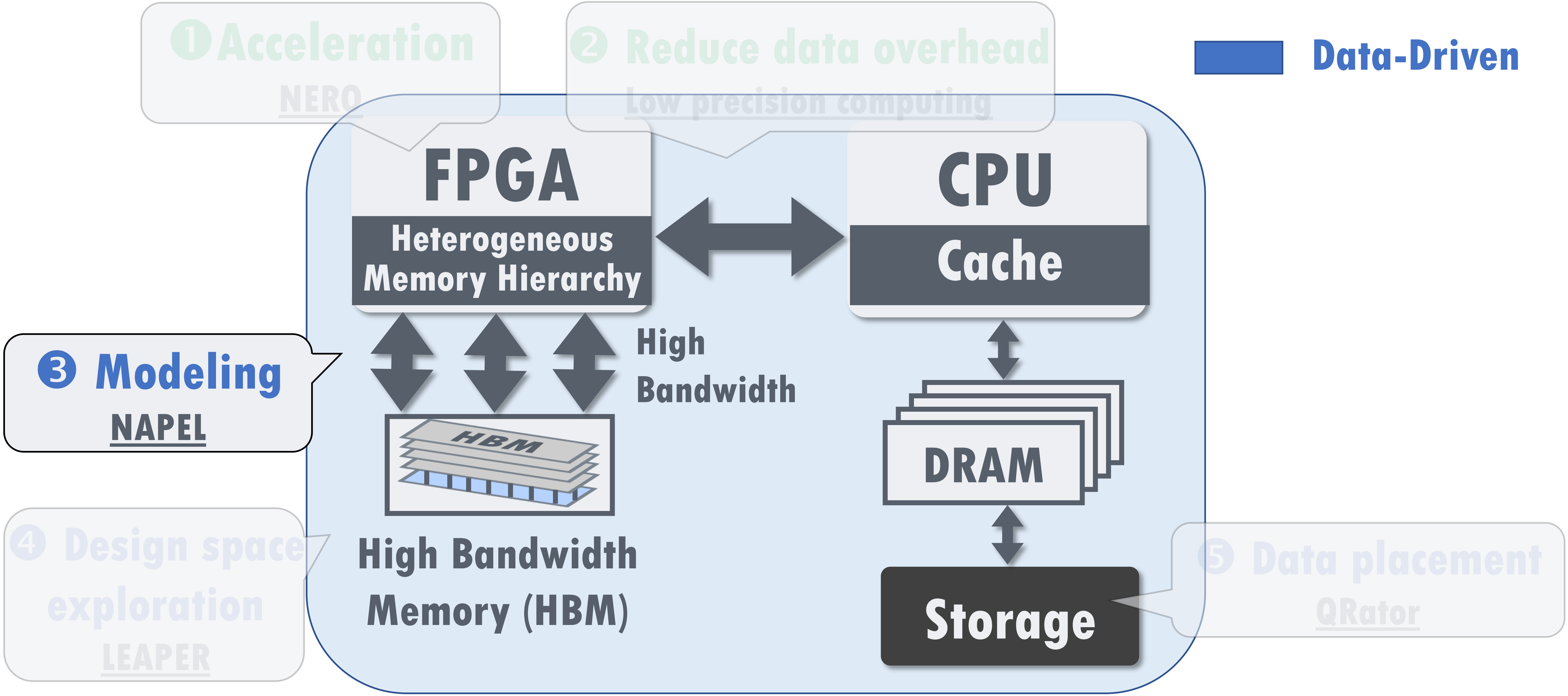


ML Model

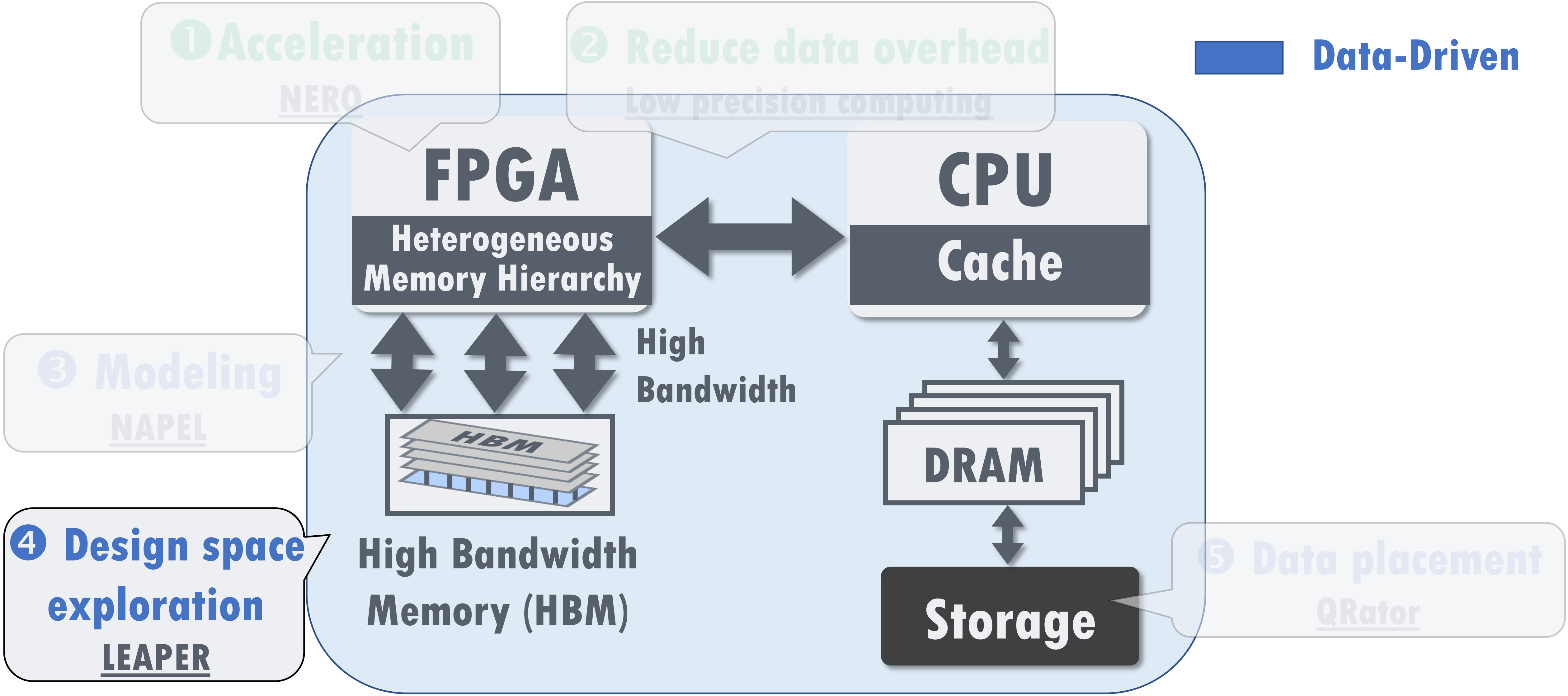
Performance/ Energy Prediction

up to 1039x faster than simulator

(3) NAPEL: ML-Based Simulation



(4) LEAPER: ML-Based DSE Framework For FPGAs

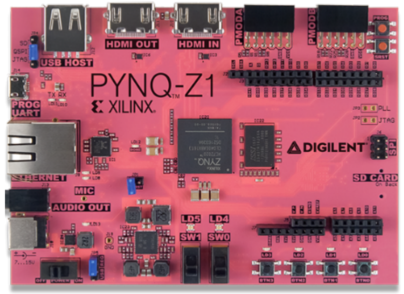


(4) LEAPER: ML-Based DSE Framework For FPGAs

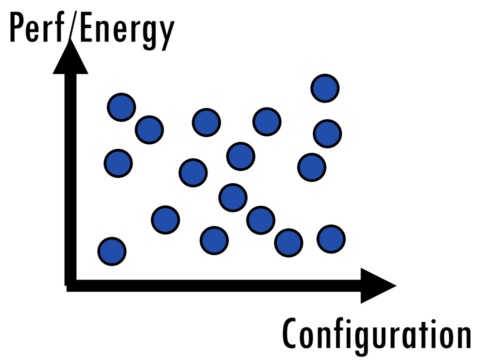
Exploration on an FPGA

Application

Synthesis
Place & Route
(~ hours)



Low-end FPGA



Design Space

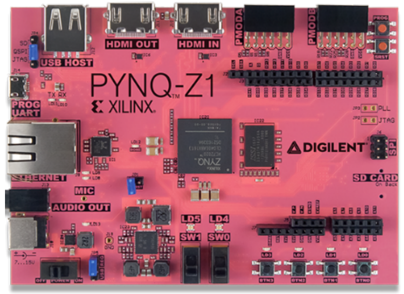
(4) LEAPER: ML-Based DSE Framework For FPGAs

Exploration on an FPGA

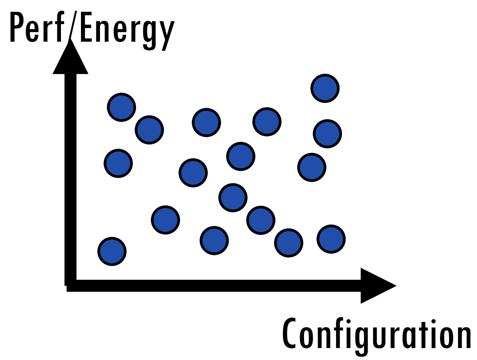
Application

VIVADO

Synthesis
Place & Route
(~ hours)



Low-end FPGA



Design Space

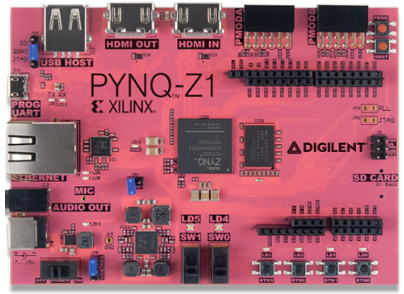
Huge design space with
time-consuming FPGA design cycle

(4) LEAPER: ML-Based DSE Framework For FPGAs

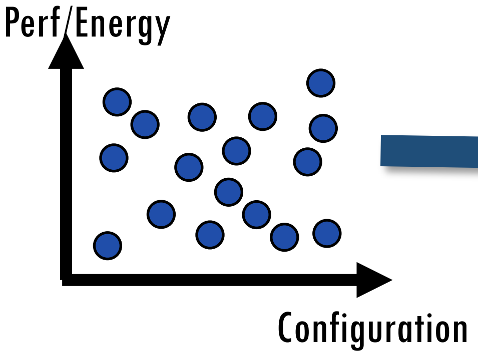
Exploration on an FPGA

Application

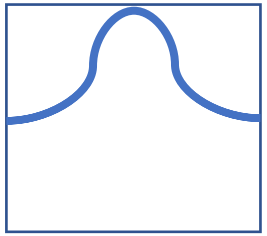
Synthesis
Place & Route
(~ hours)



Low-end FPGA



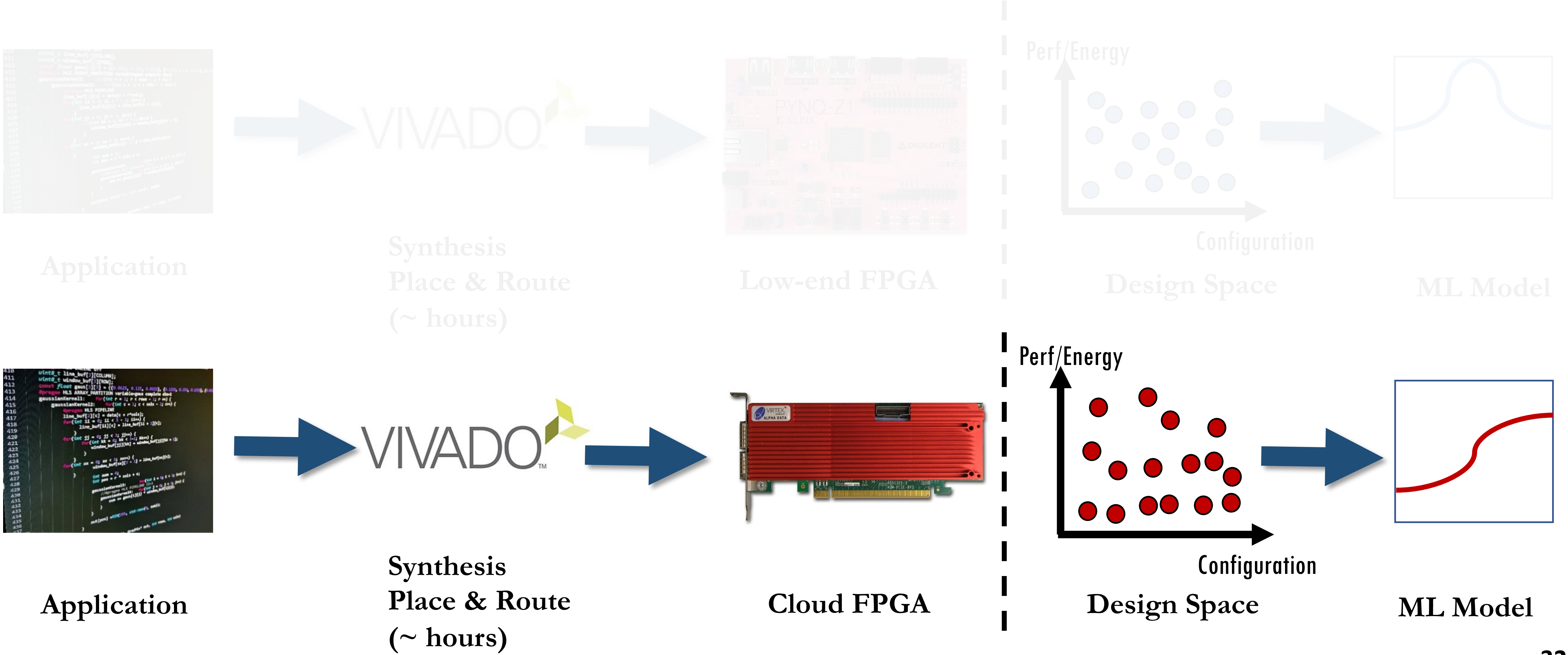
Design Space



ML Model

(4) LEAPER: ML-Based DSE Framework For FPGAs

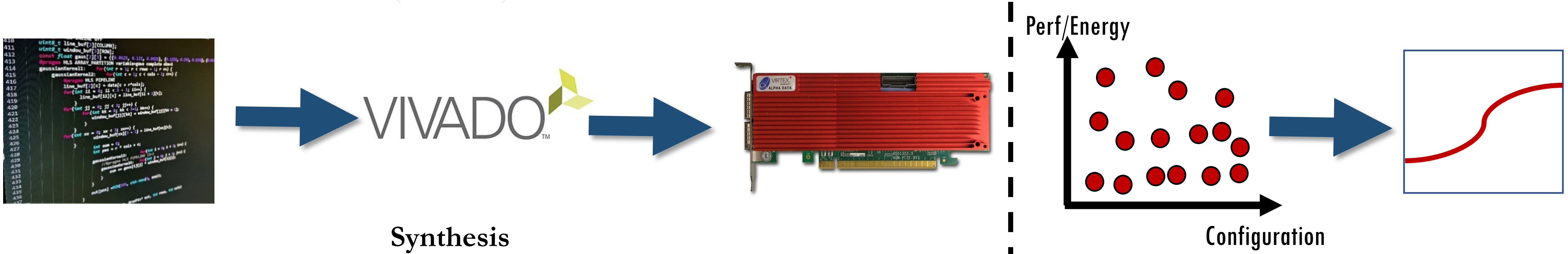
Exploration on a Different FPGA



(4) LEAPER: ML-Based DSE Framework For FPGAs

Exploration on a Different FPGA

Model trained for a specific environment cannot predict for a new, unknown environment



Application

Synthesis
Place & Route
(~ hours)

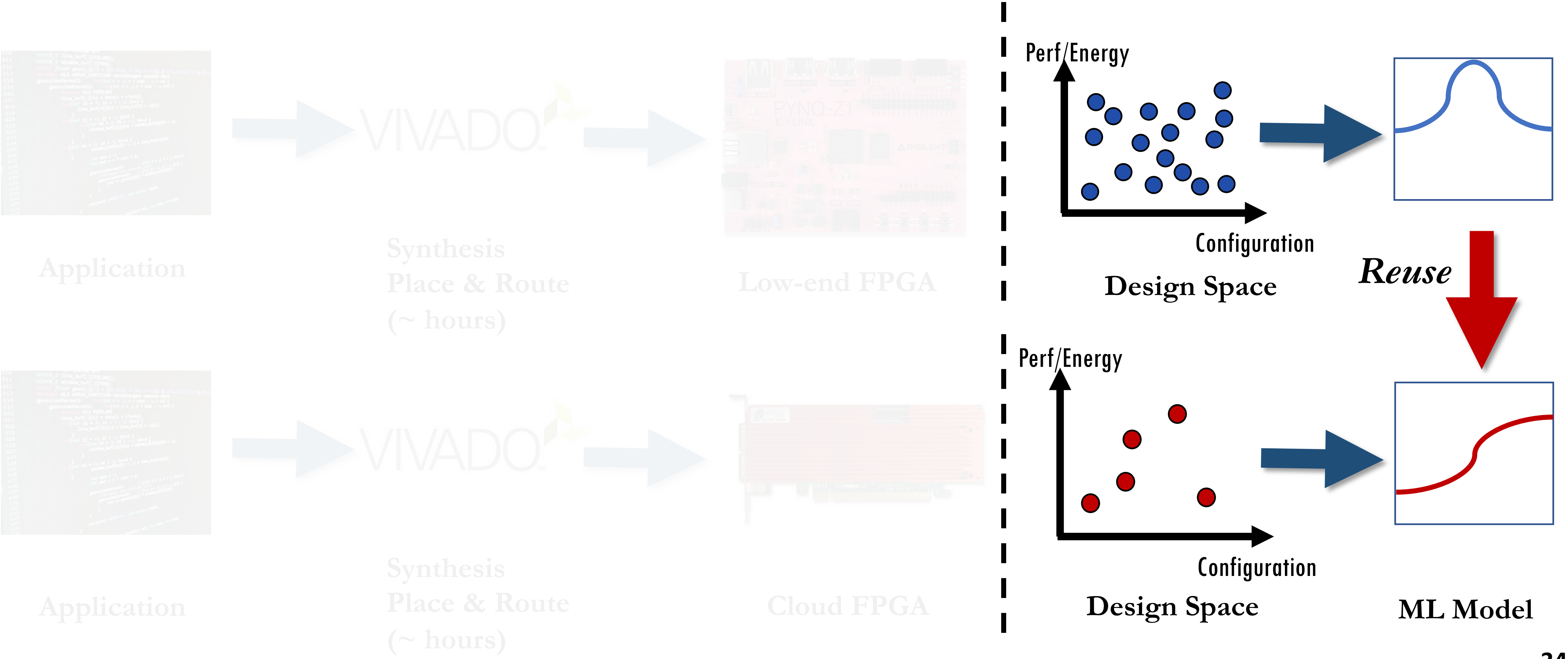
Cloud FPGA

Design Space

ML Model

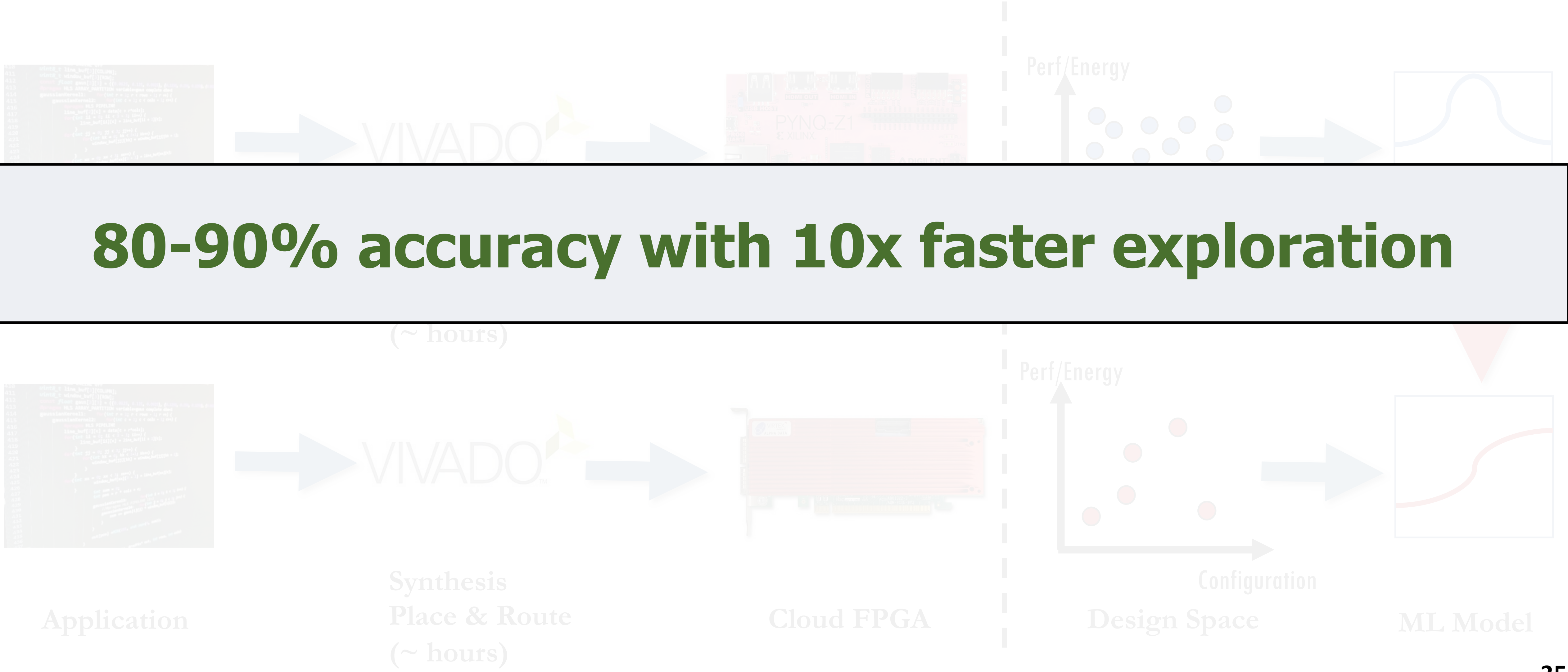
(4) LEAPER: ML-Based DSE Framework For FPGAs

Exploration on a Different FPGA



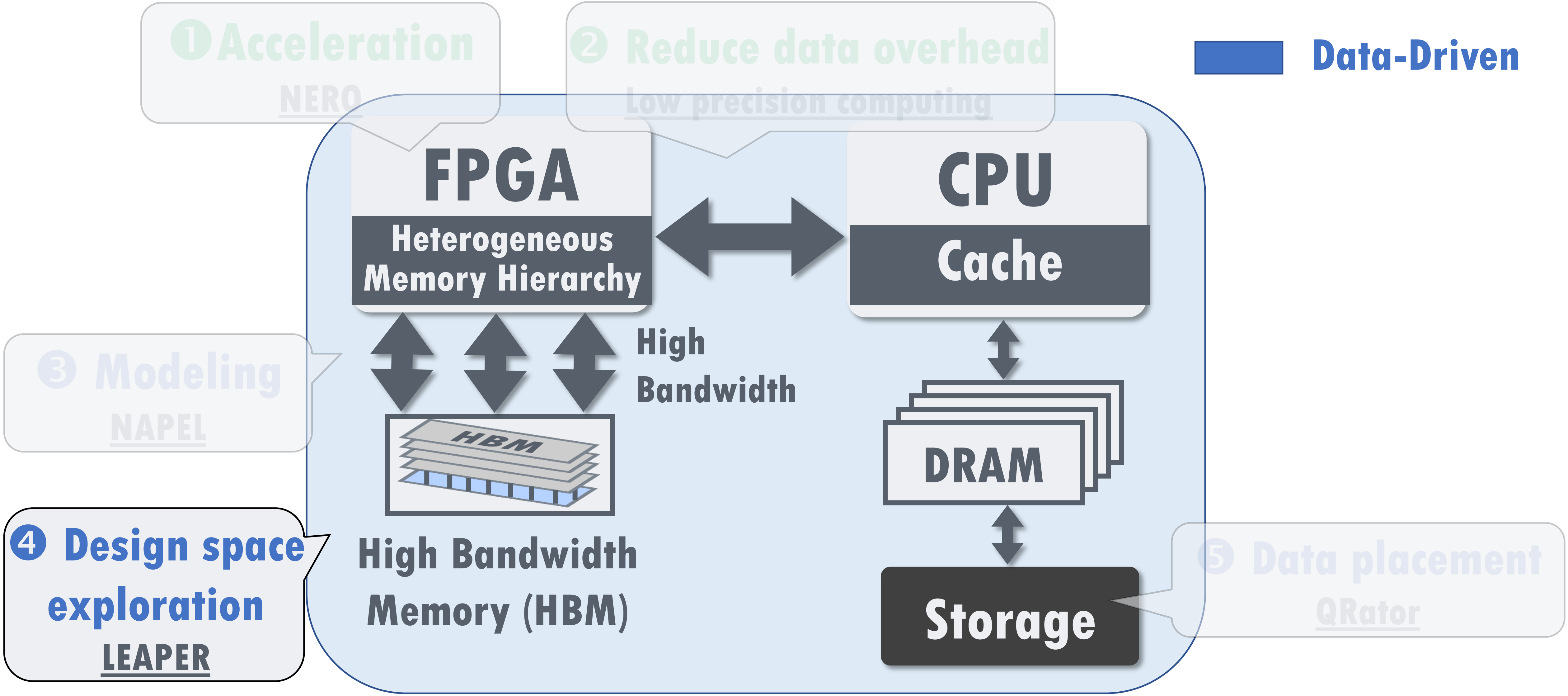
(4) LEAPER: ML-Based DSE Framework For FPGAs

Exploration on a Different FPGA

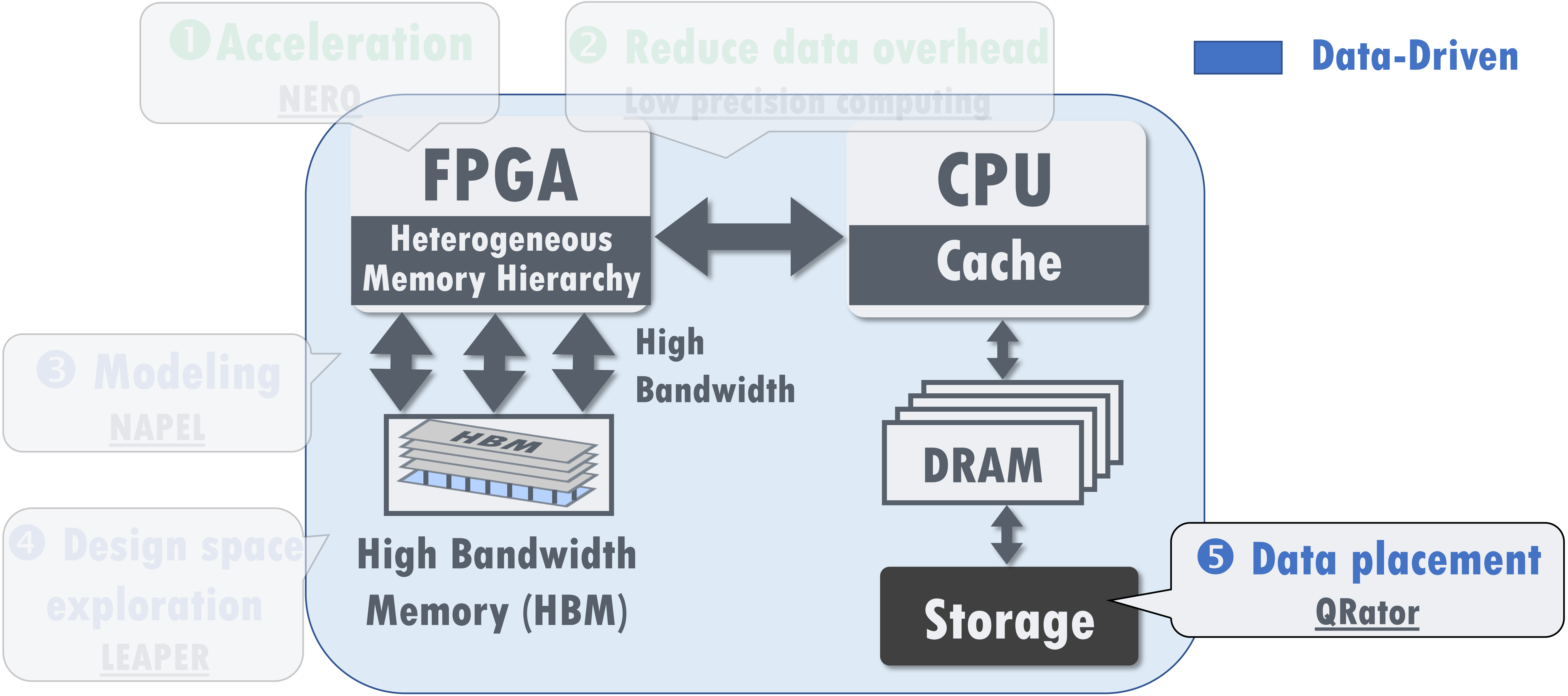


80-90% accuracy with 10x faster exploration

(4) LEAPER: ML-Based DSE Framework For FPGAs

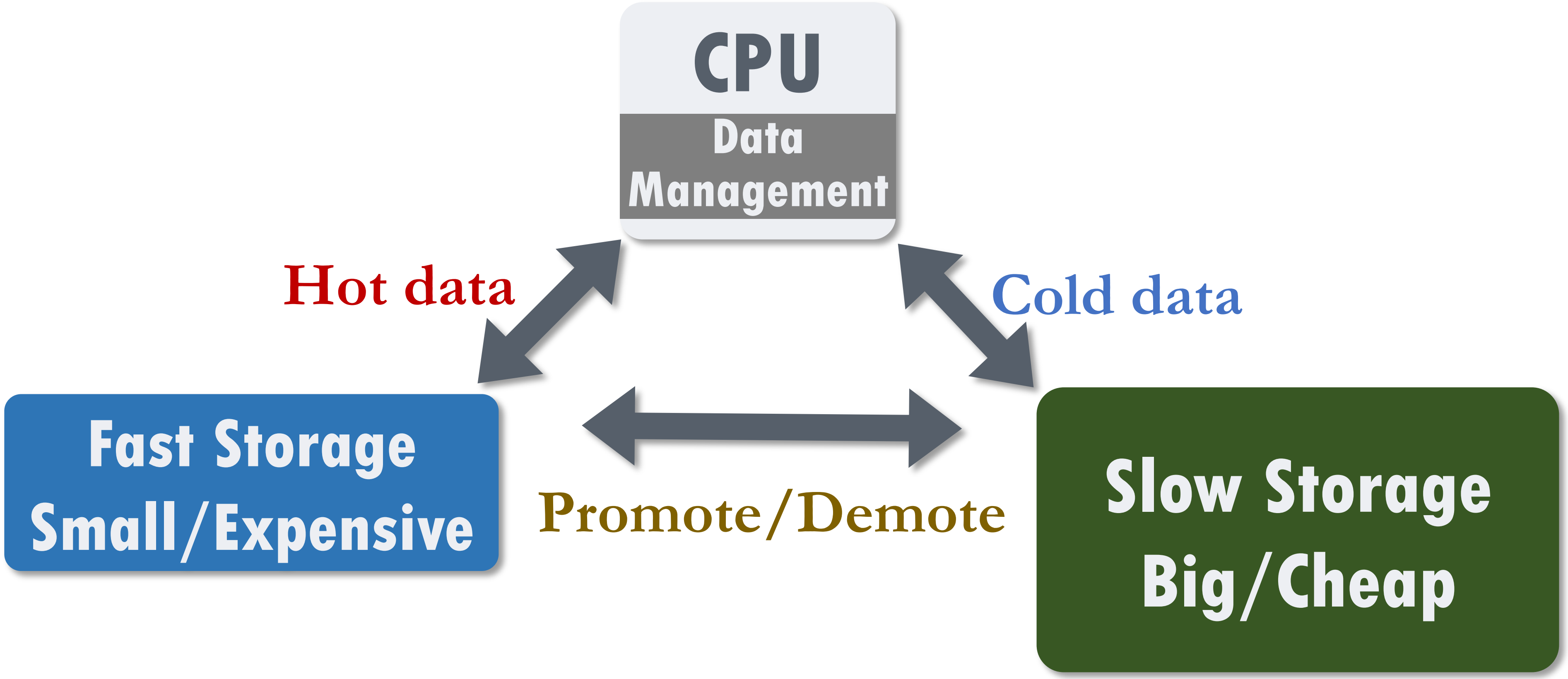


(5) QRator: Efficient Data-Placement Mechanism



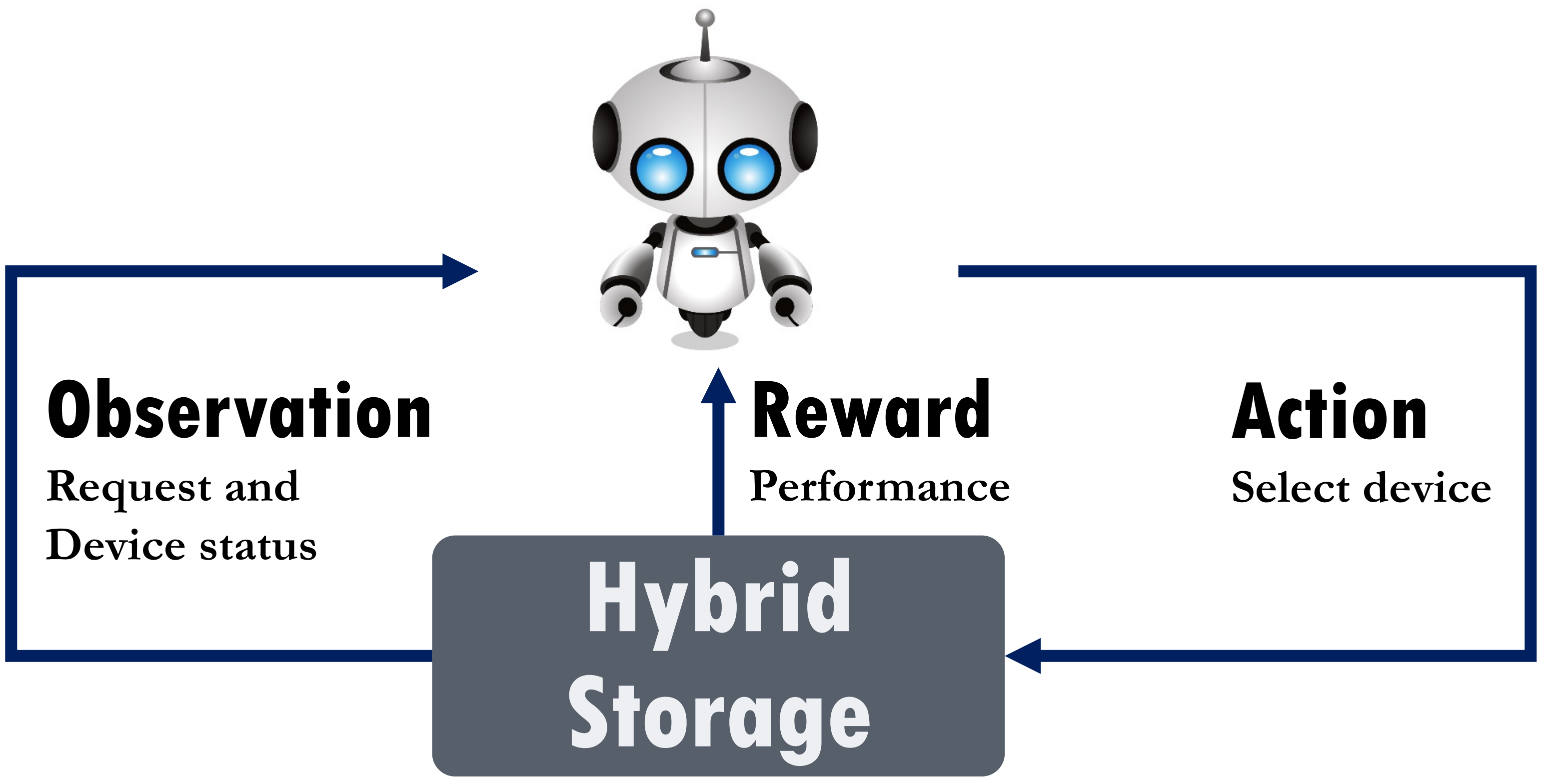
(5) QRator: Efficient Data-Placement Mechanism

Hybrid Storage Subsystem



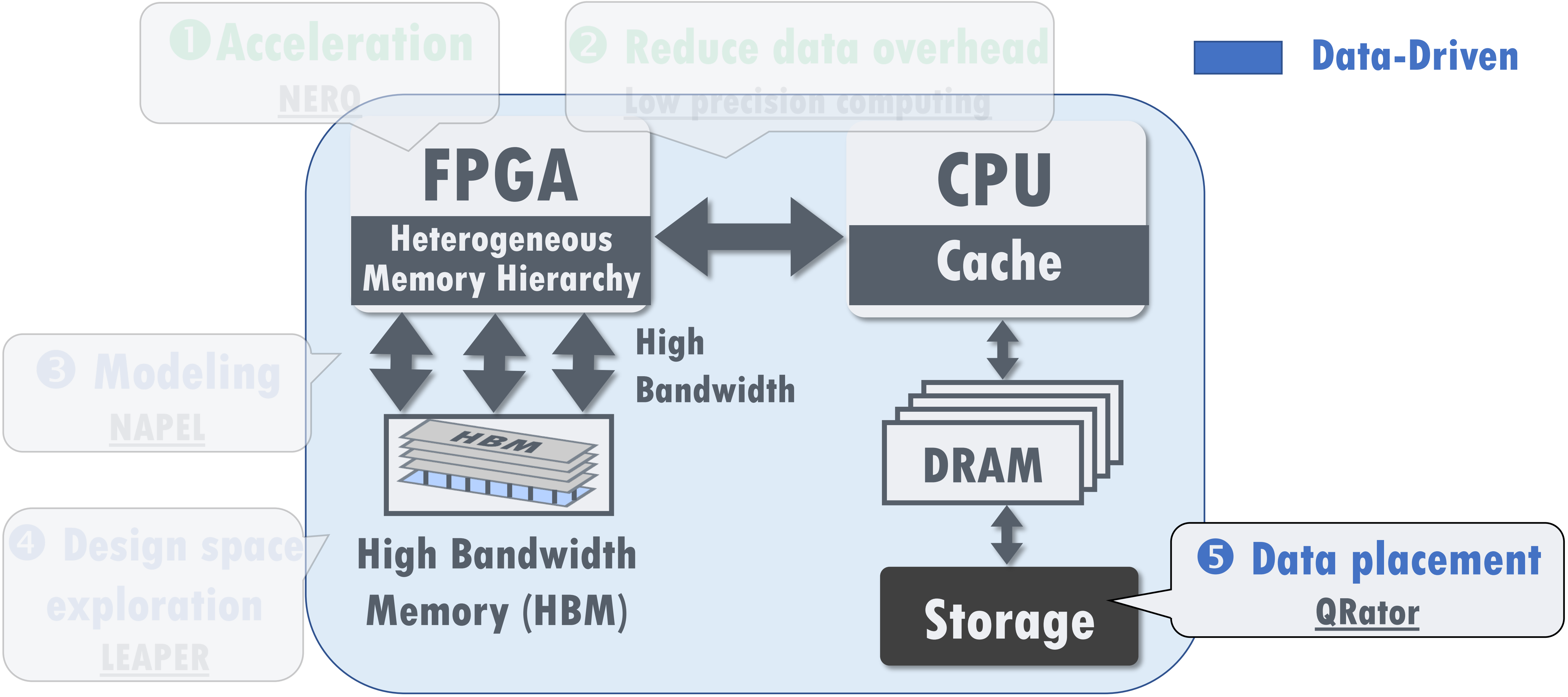
Self-adaptable, efficient data-placement is challenging

(5) QRator: Efficient Data-Placement Mechanism

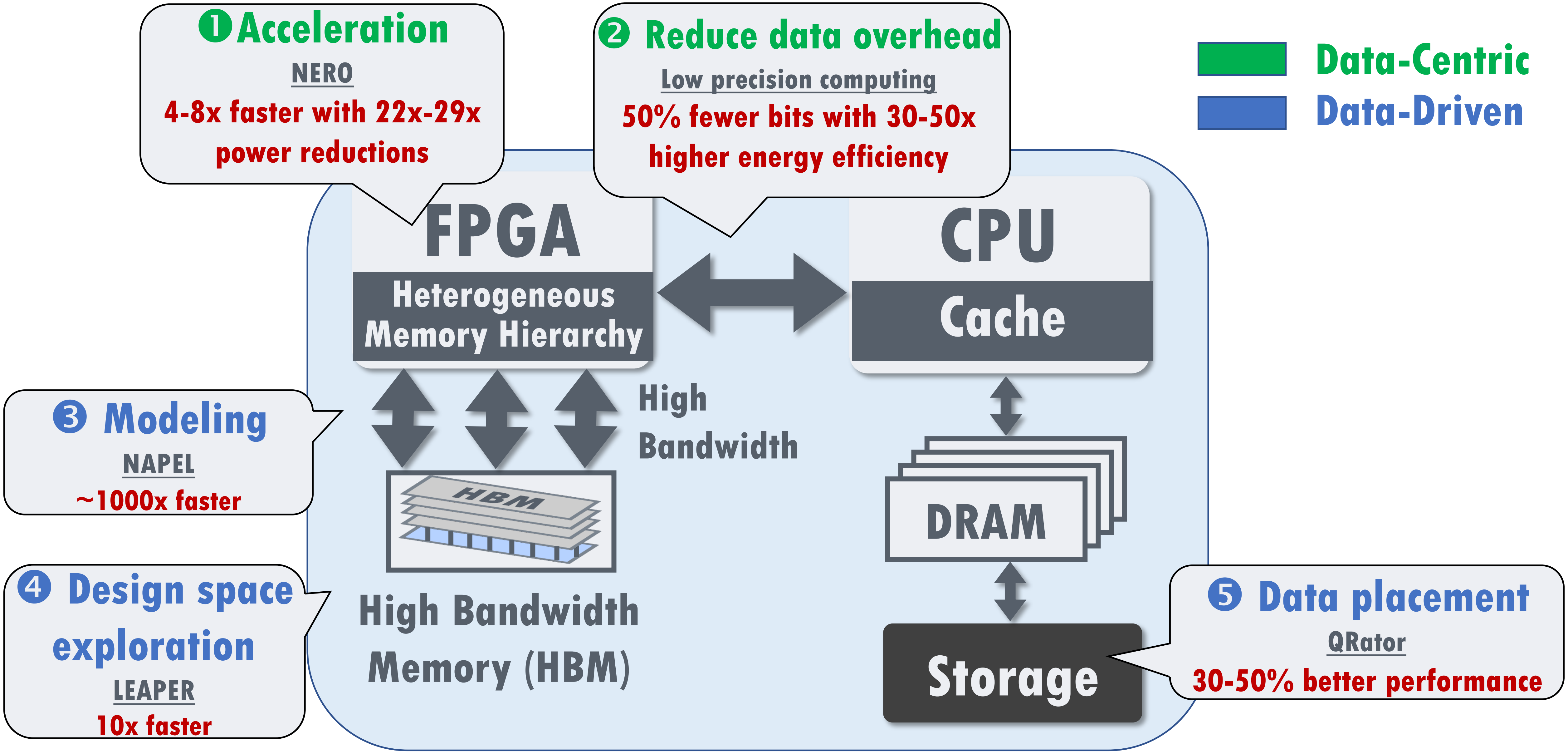


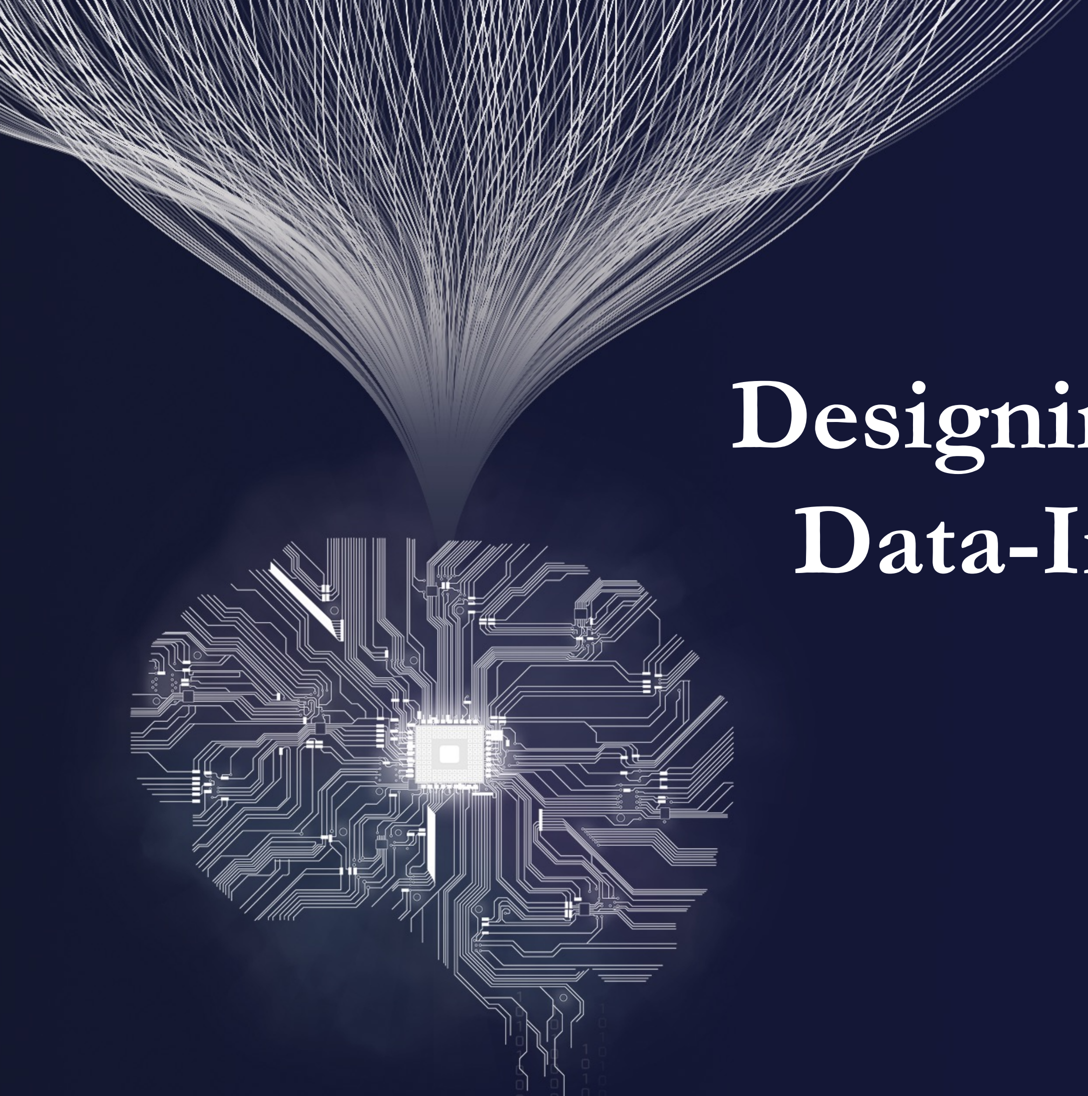
Performance improvement of **30-50%** compared to **state-of-the-art data-placement techniques**

(5) QRator: Efficient Data-Placement Mechanism



Thesis Contributions





SAFARI

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Designing, Modeling, and Optimizing Data-Intensive Computing Systems

Gagandeep Singh

Ph.D. Defense

Committee:

Henk Corporaal (TU Eindhoven)

Onur Mutlu (ETH, Zurich)

Sander Stuijk (TU Eindhoven)

C.H. Berkel (TU Eindhoven)

Peter Hofstee (IBM Austin/TU Delft)

Francky Catthoor (IMEC/KU Leuven)

Dionysios Diamantapoulos (IBM Research Europe)

Osman Unsal (BSC)

Backup

NERO:
A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling

Executive Summary

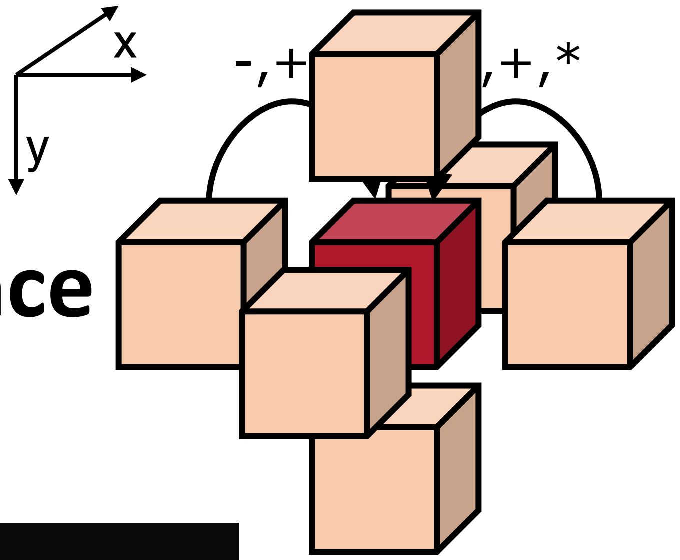
- **Motivation:** Stencil computation is an essential part of weather prediction applications
- **Problem:** Memory bound with limited performance and high energy consumption on multi-core architectures
- **Goal:** Mitigate the performance bottleneck of compound weather prediction kernels in an energy-efficient way
- **Our contribution: NERO**
  - First near High-Bandwidth Memory (HBM) FPGA-based accelerator for representative kernels from a real-world weather prediction application
  - Detailed roofline analysis to show weather prediction kernels are constrained by DRAM bandwidth on a state-of-the-art CPU system
  - Data-centric caching with precision-optimized tiling for a heterogeneous memory hierarchy
  - Scalability analysis for both DDR4 and HBM-based FPGA boards
- **Evaluation**
  - NERO outperforms a 16-core IBM POWER9 system by 4.2x and 8.3x when running two compound stencil kernels
  - NERO reduces energy consumption by 22x and 29x with an energy efficiency of 1.5 GFLOPS/Watt and 17.3 GFLOPS/Watt

Outline

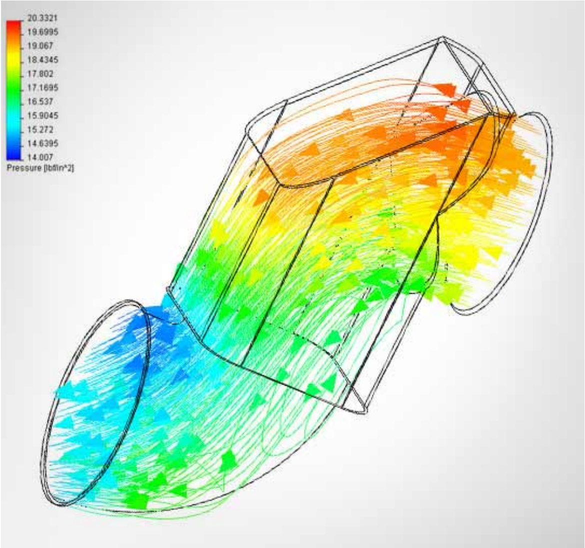
Stencil Computations and Applications

Stencil computations update values in a grid using a **fixed pattern** of grid points

Stencils are used in **~30% of high-performance computing applications**



e.g., 7-point Jacobi in 3D plane



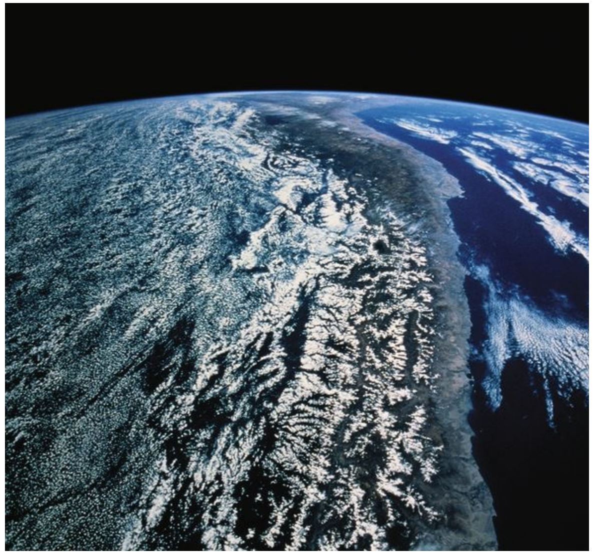
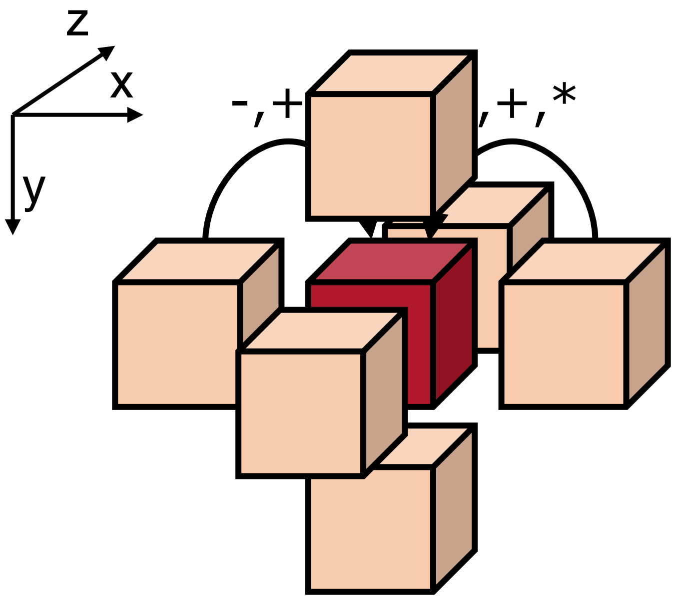


Image sources: http://www.flometrics.com/fluid-dynamics/computational-fluid-dynamics
Naoe, Kensuke et al. "Secure Key Generation for Static Visual Watermarking by Machine Learning in Intelligent Systems and Services" IJSSOE, 2010

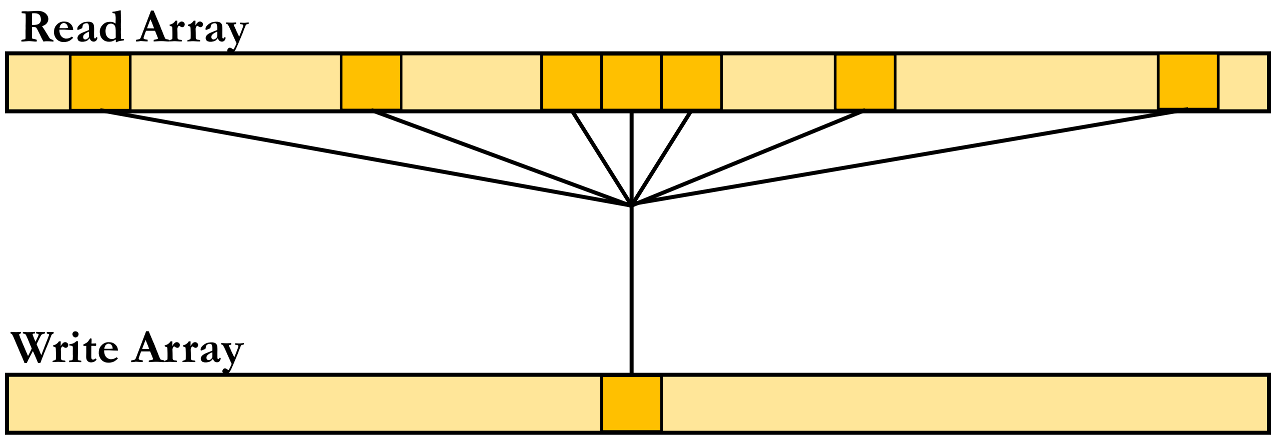
Stencil Characteristics

High-order stencil computations are cache unfriendly

- Limited arithmetic intensity
- Sparse and complex access pattern



e.g., 7-point Jacobi in 3D plane



Mapping of 7-point Jacobi from 3D plane onto 1D plane

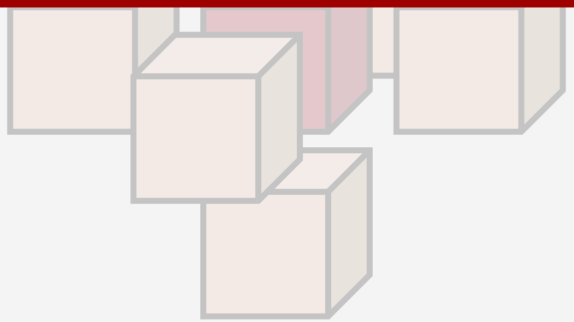
Image source: Xu, Jingheng et al. "Performance Tuning and Analysis for Stencil-Based Applications on POWER8 Processor" ACM TACO, 2018

Stencil Characteristics

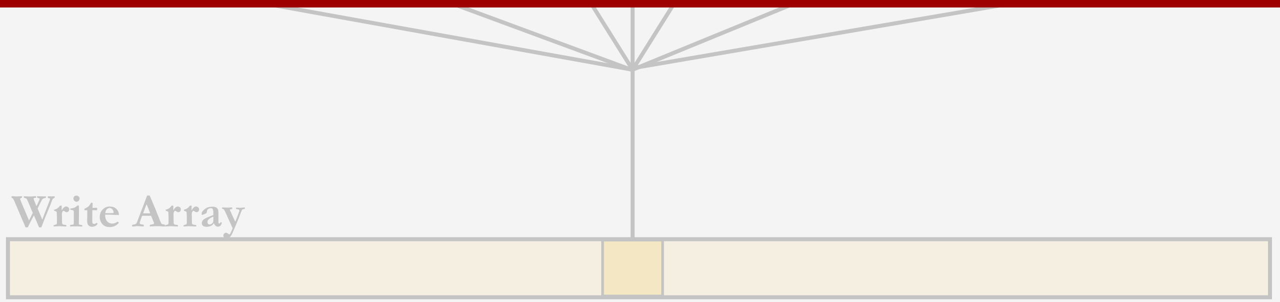
High-order stencil computations are cache unfriendly

- Limited arithmetic intensity

Performance bottleneck



e.g., 7-point Jacobi in 3D plane



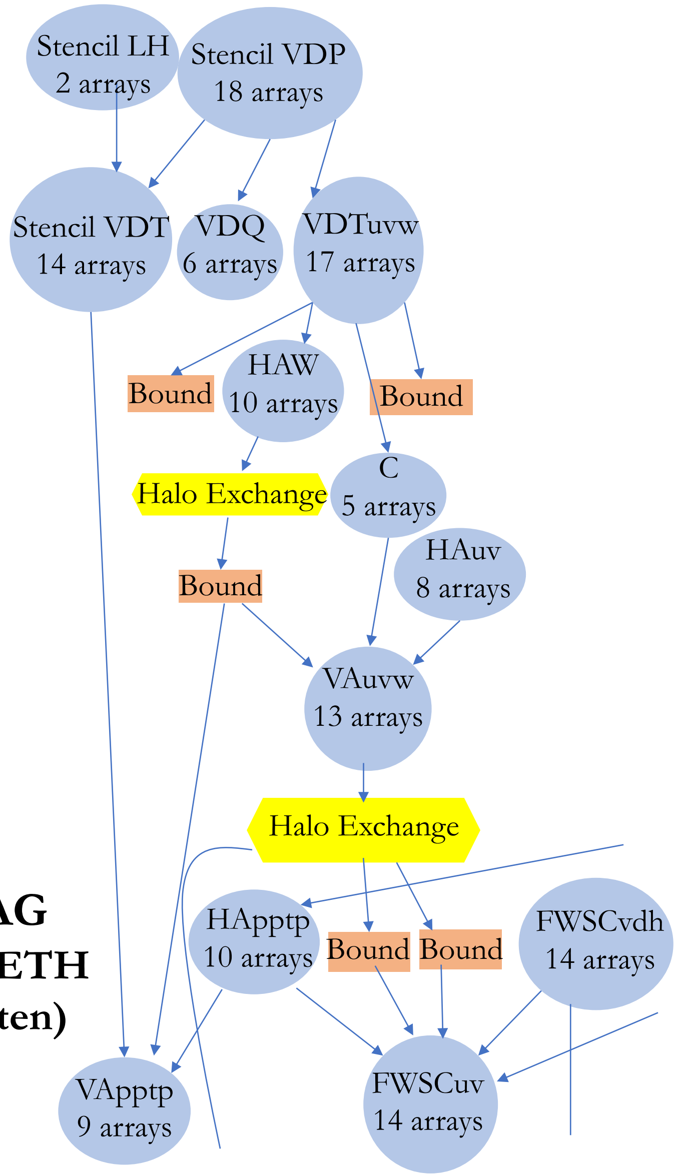
Mapping of 7-point Jacobi from 3D plane onto 1D plane

Image source: Xu, Jingheng et al. "Performance Tuning and Analysis for Stencil-Based Applications on POWER8 Processor" ACM TACO, 2018

Stencil Computations in Weather Applications

COSMO (Consortium for Small-Scale Modeling) weather prediction application

- The essential part of the weather prediction models is called **dynamical core**
- Around **80 different** stencil compute motifs
- ~30 variables and ~70 temporary arrays (3D grids)
- Horizontal diffusion and vertical advection
- **Complex stencil programs**



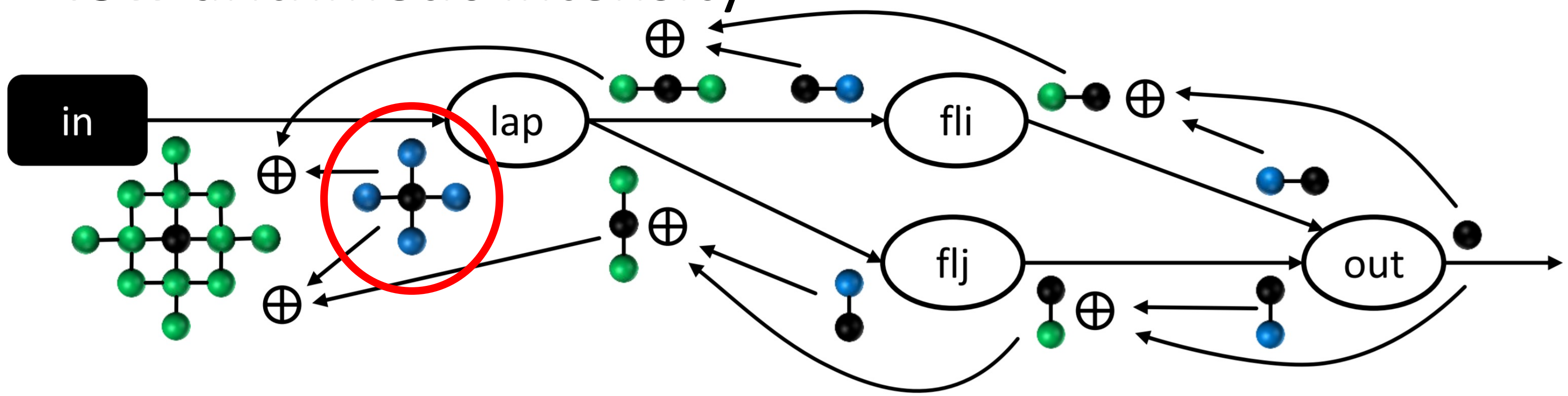
Section of COSMO CDAG (Courtesy CSCS/ETH and Ronald Luijten)

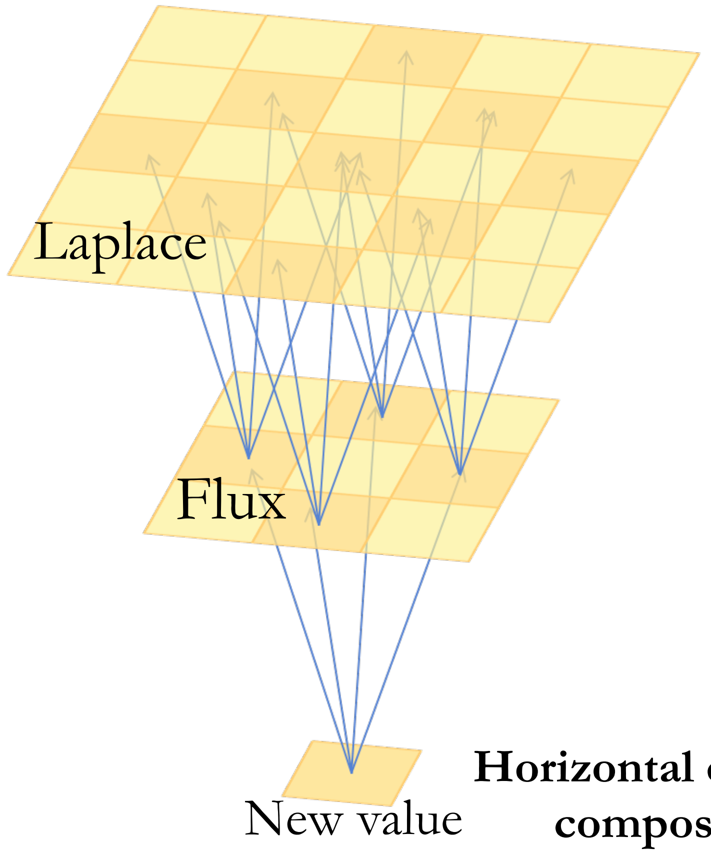
Example Complex Stencil: Horizontal Diffusion

- Compound stencil kernel consists of a **collection** of elementary stencil kernels

- Iterates over a 3D grid performing **Laplacian** and **flux** operations

- **Complex** memory access behavior and **low** arithmetic intensity





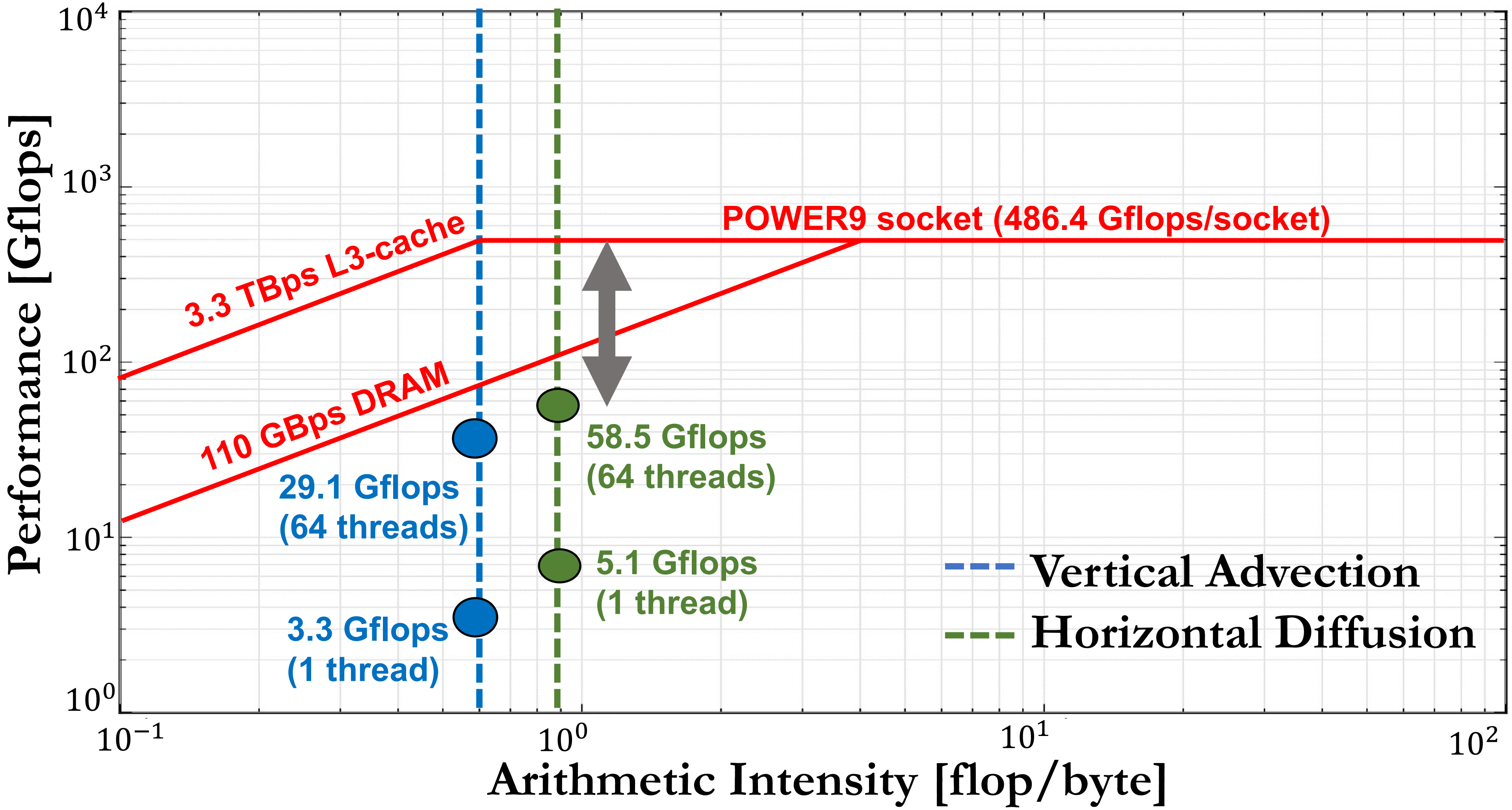
Horizontal diffusion composition (Courtesy CSCS/ETH and Ronald Luijten)

DFG source: Gysi, Tobias, et al. "MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures" ICS, 2015

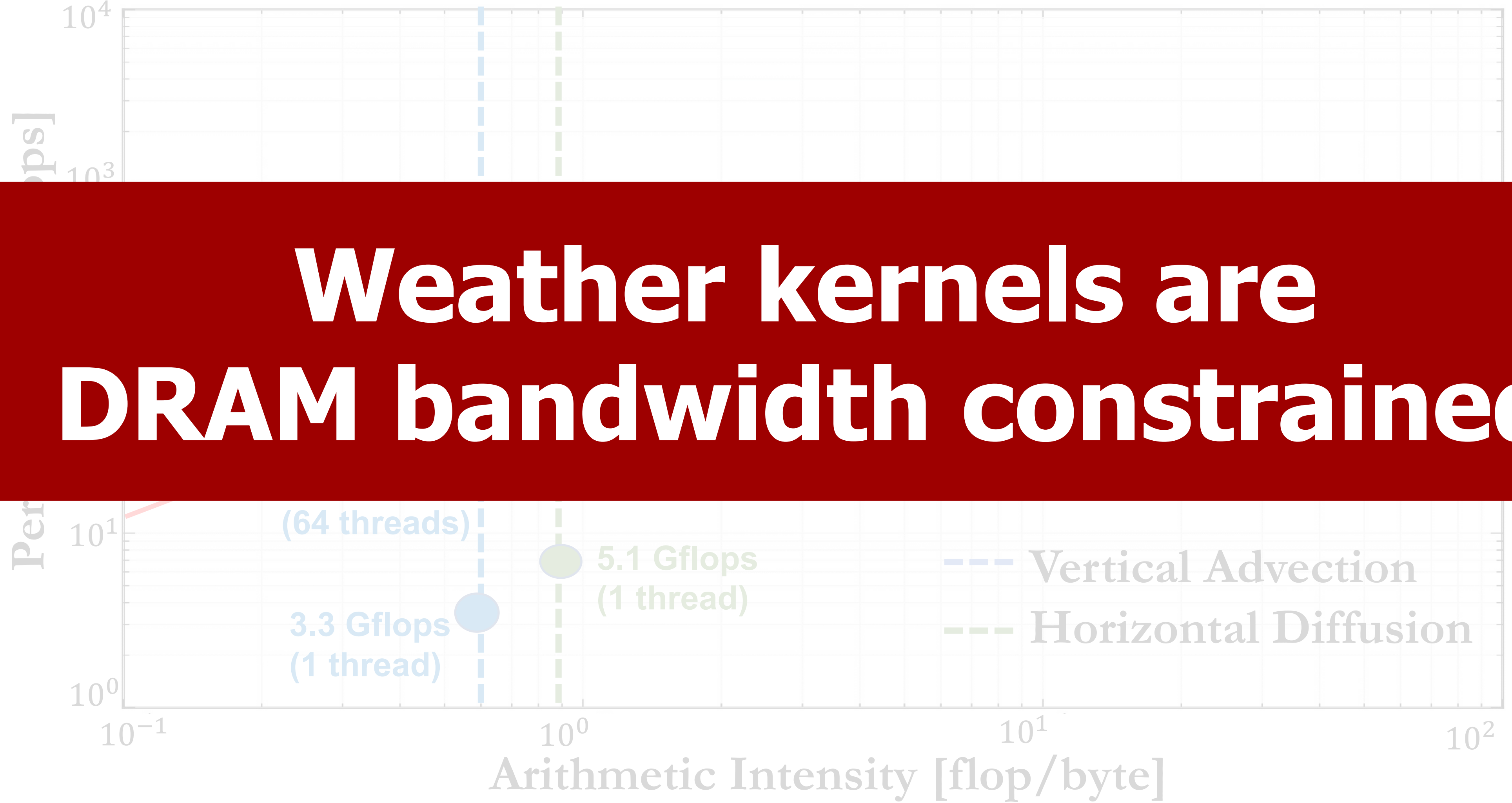
Outline

- Background
- **CPU Roofline Analysis**
- FPGA-based Platform
- NERO: Near-HBM Accelerator for Weather Prediction Modeling
- Precision-optimized Tiling
- Evaluation
  - Performance Analysis
  - Energy Efficiency
- Summary

IBM POWER9 Roofline Analysis



IBM POWER9 Roofline Analysis



Weather kernels are DRAM bandwidth constrained

Outline

Background

CPU Roofline Analysis

FPGA-based Platform

NERO: **Near-HBM Accelerator for Weather Prediction Modeling**

Precision-optimized Tiling

Evaluation

- Performance Analysis
- Energy Efficiency Analysis

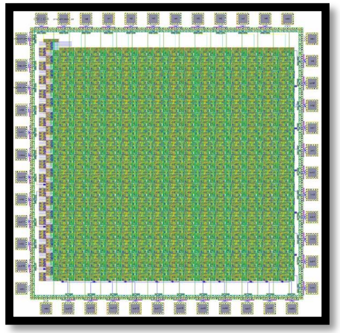
Summary

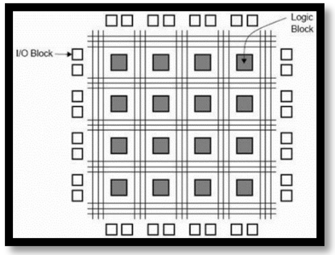
Silicon Alternatives

CPUs

GPUs

FPGAs





ASICs

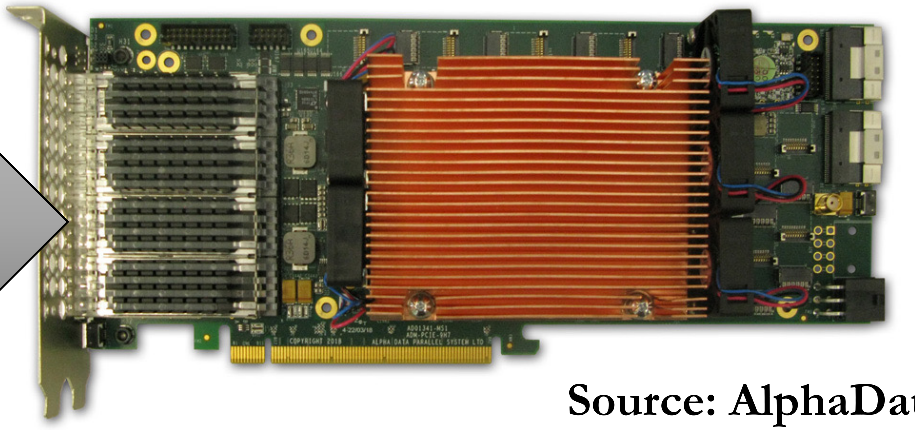
FLEXIBILITY **EFFICIENCY**

FPGAs are highly configurable!

Heterogeneous System: CPU+FPGA

Source: IBM

POWER9 AC922



Source: AlphaData

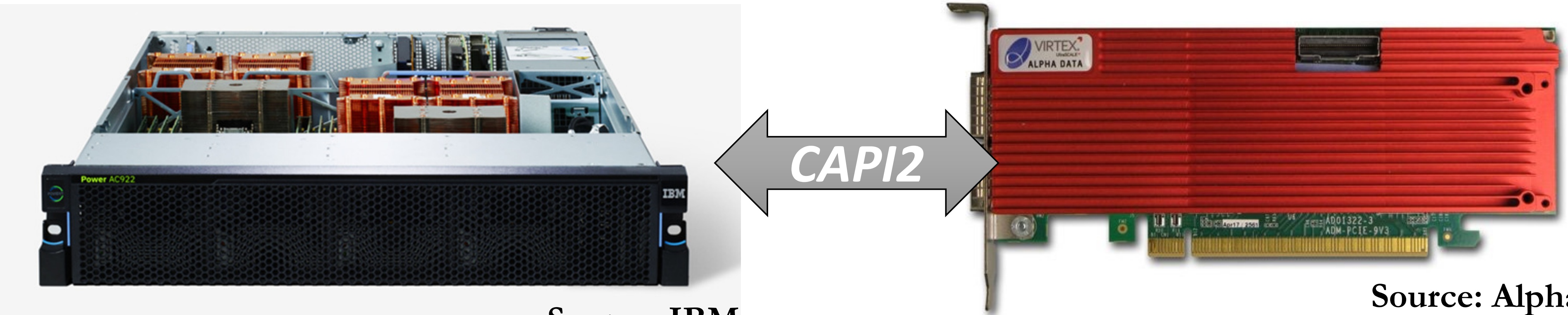
HBM-based AD9H7 board

We evaluate two POWER9+FPGA systems:

1. HBM-based board AD9H7

Xilinx Virtex Ultrascale+™ XCVU37P-2

Heterogeneous System: CPU+FPGA



Source: IBM

Source: AlphaData

POWER9 AC922

DDR4-based AD9V3 board

We evaluate two POWER9+FPGA systems:

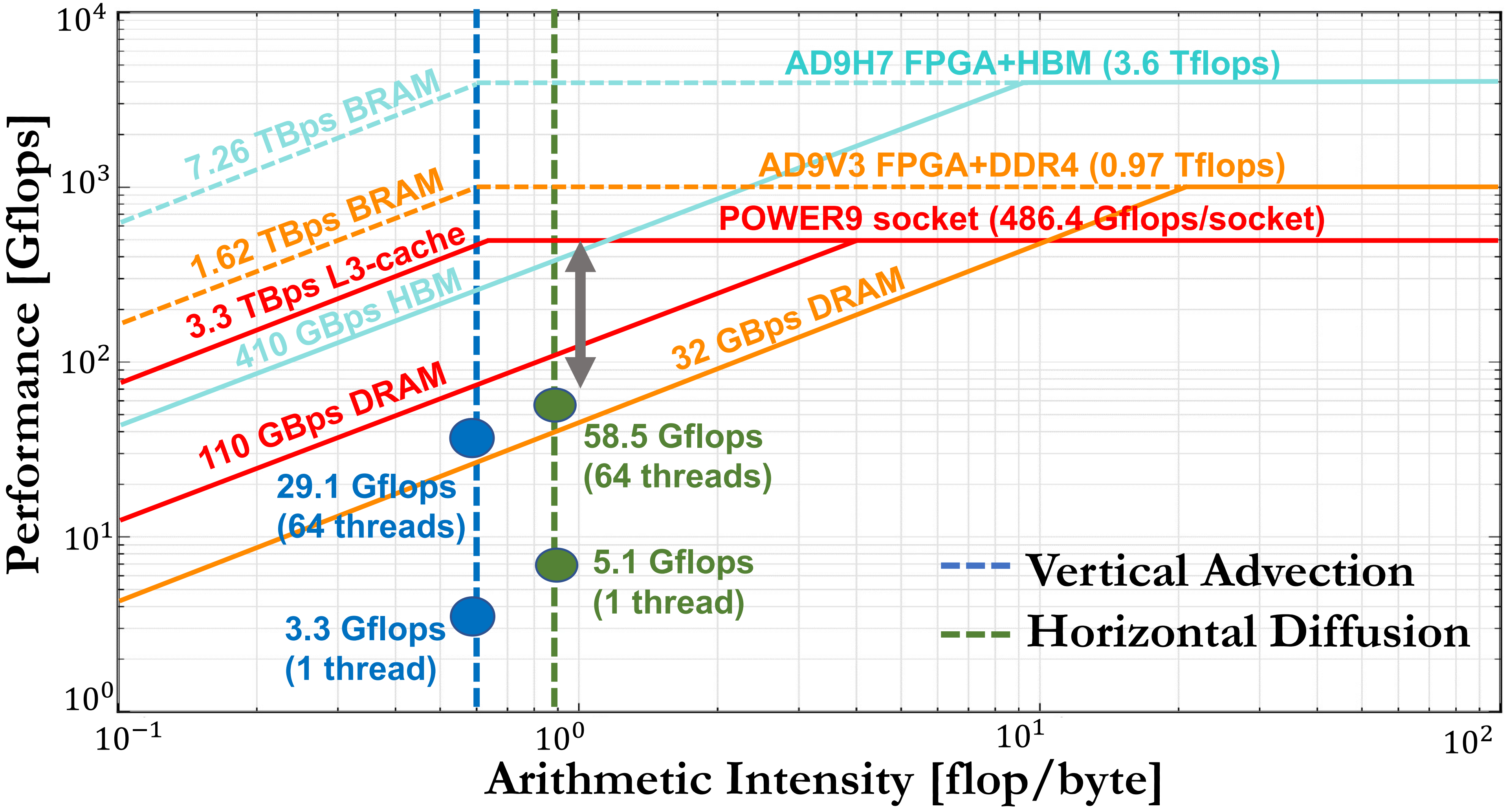
1. HBM-based board AD9H7

Xilinx Virtex Ultrascale+™ XCVU37P-2

2. DDR4-based board AD9V3

Xilinx Virtex Ultrascale+™ XCVU3P-2

FPGAs Have Tremendous Potential



Outline

Background

CPU Roofline Analysis

FPGA-based Platform

NERO: Near-HBM Accelerator for Weather Prediction Modeling

Precision-optimized Tiling

Evaluation

- Performance Analysis
- Energy Efficiency Analysis

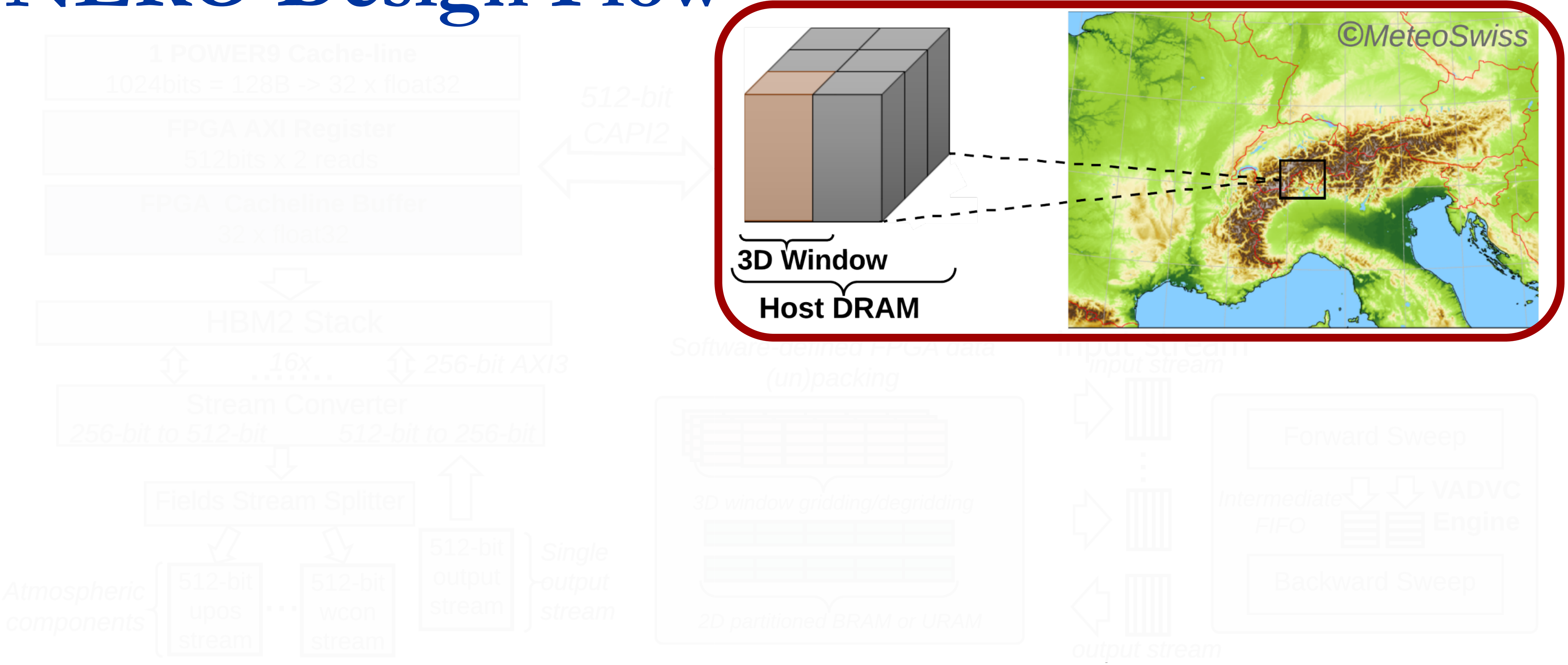
Summary

NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling

- First **near-HBM FPGA-based** accelerator for representative kernels from a **real-world weather prediction application**
- Data-centric caching with **precision-optimized tiling** for a heterogeneous memory hierarchy
- In-depth **scalability analysis** for both DDR4 and HBM-based FPGA boards

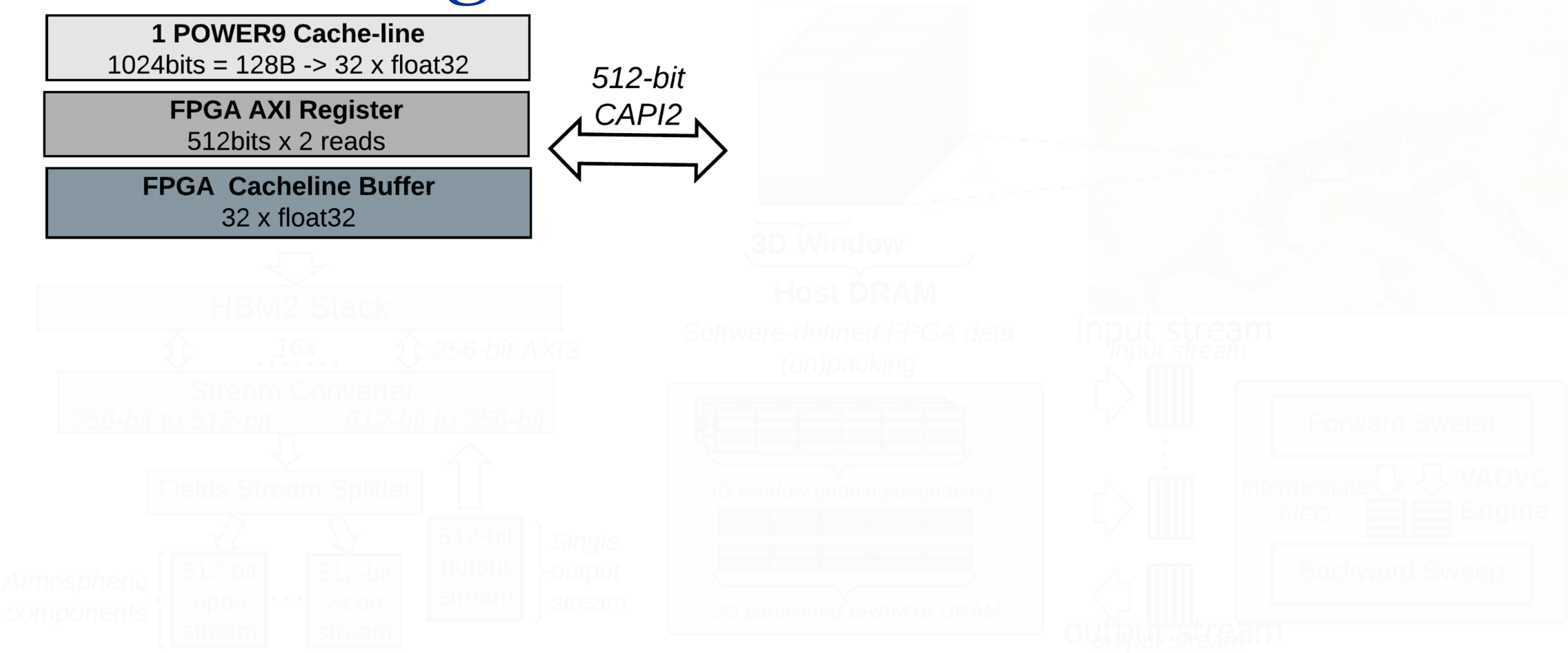
NERO Design Flow

NERO Design Flow



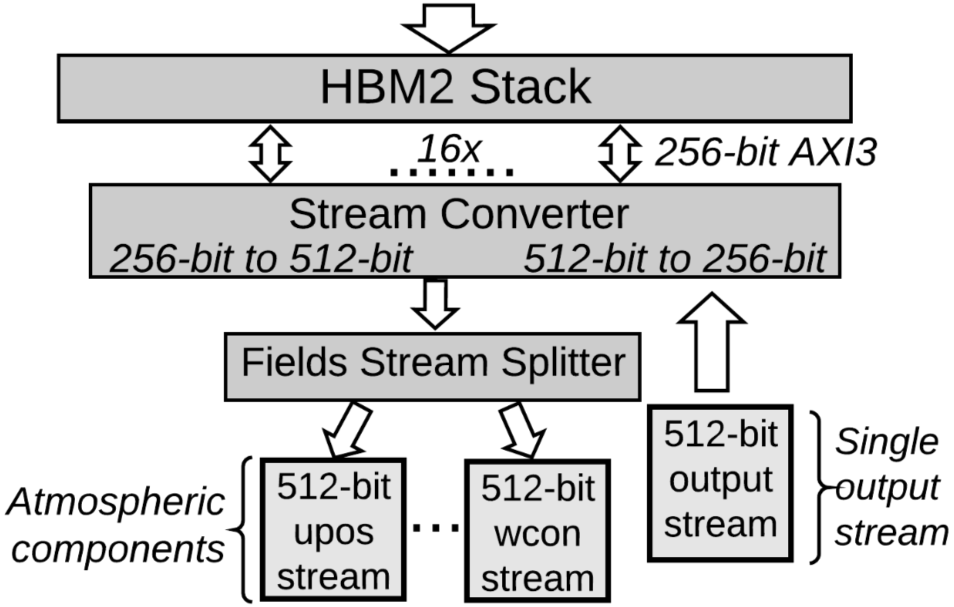
Weather data in the host DRAM

NERO Design Flow



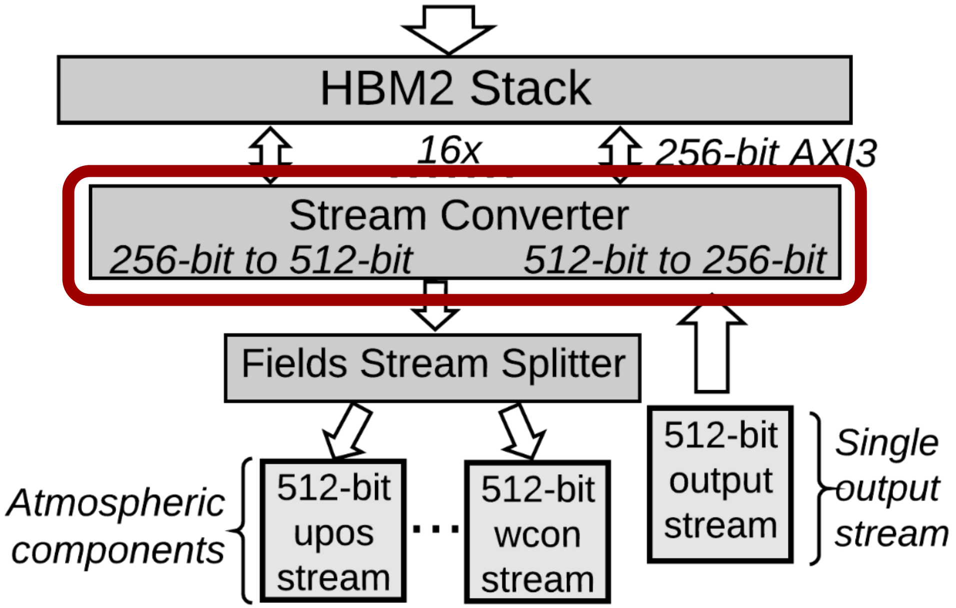
Cache-line transfer over CAPI2

NERO Design Flow



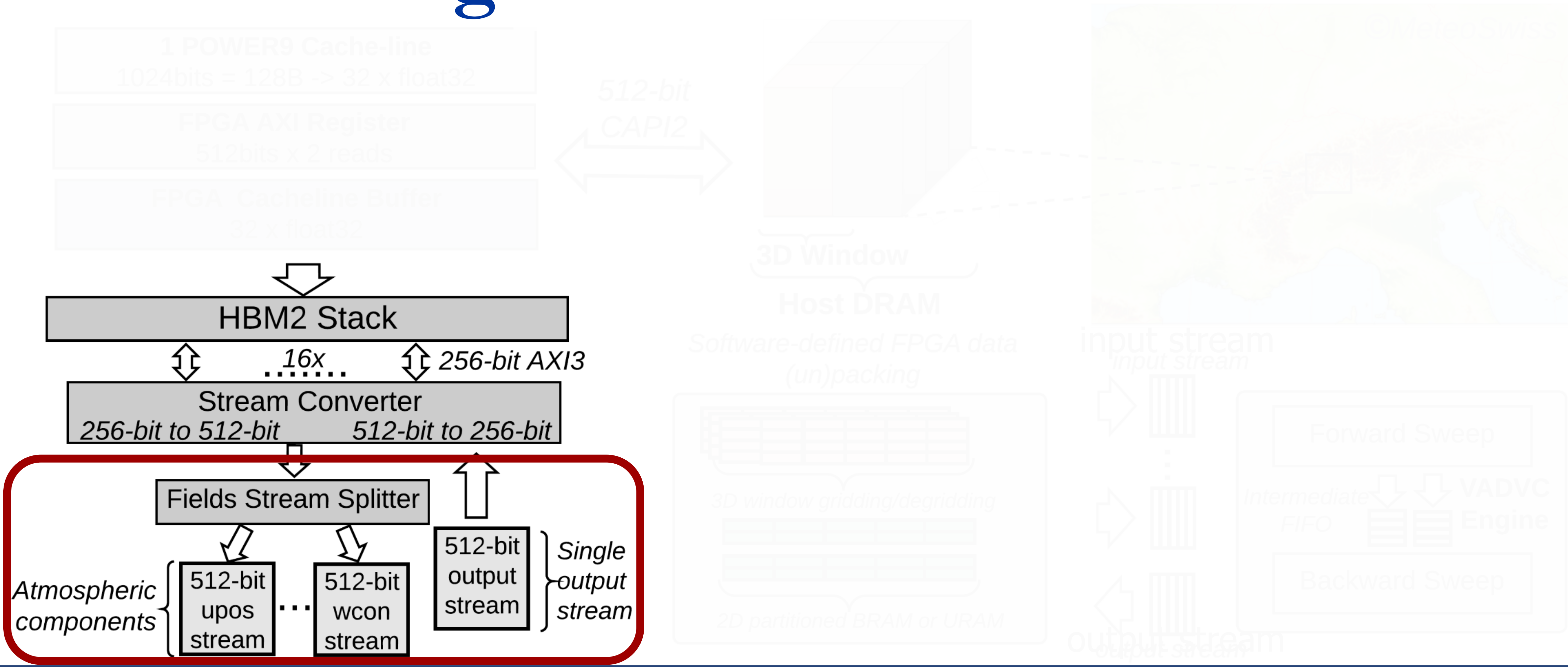
Data mapping onto HBM

NERO Design Flow



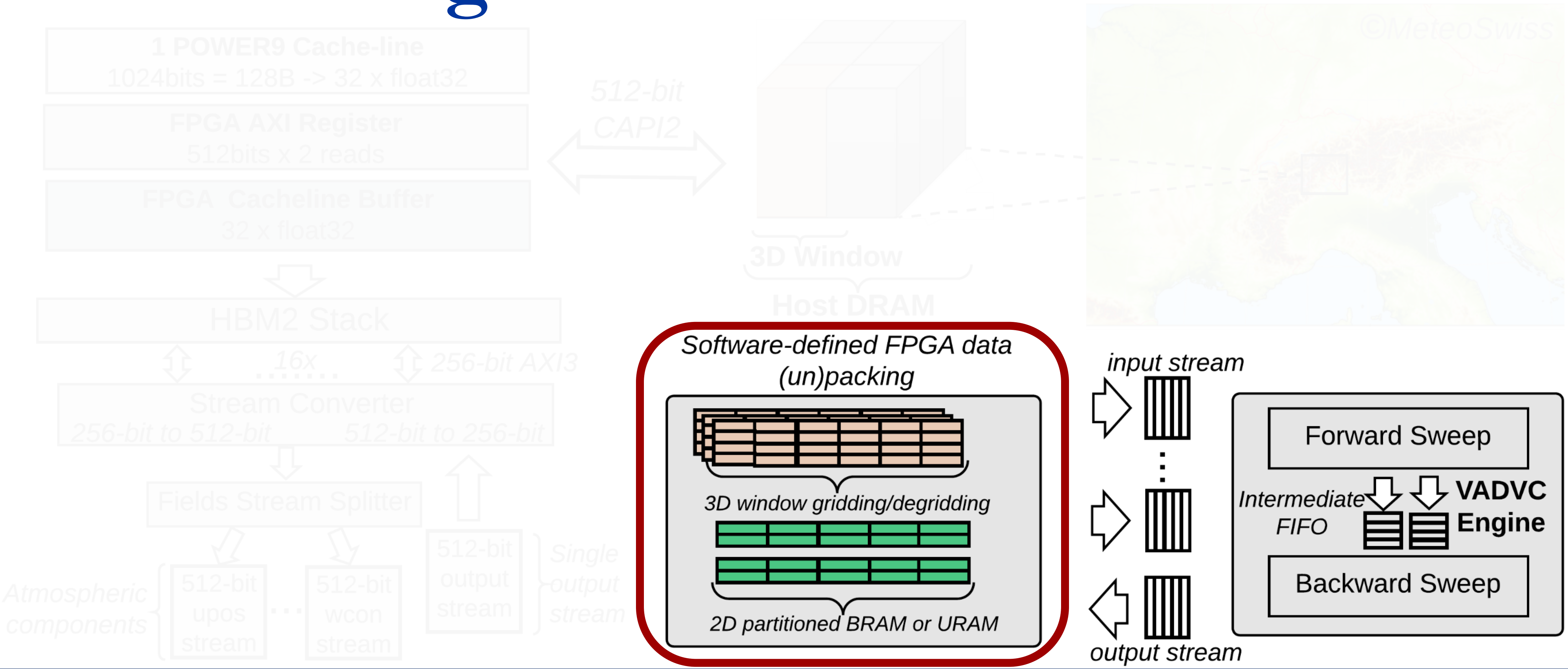
Data mapping onto HBM

NERO Design Flow



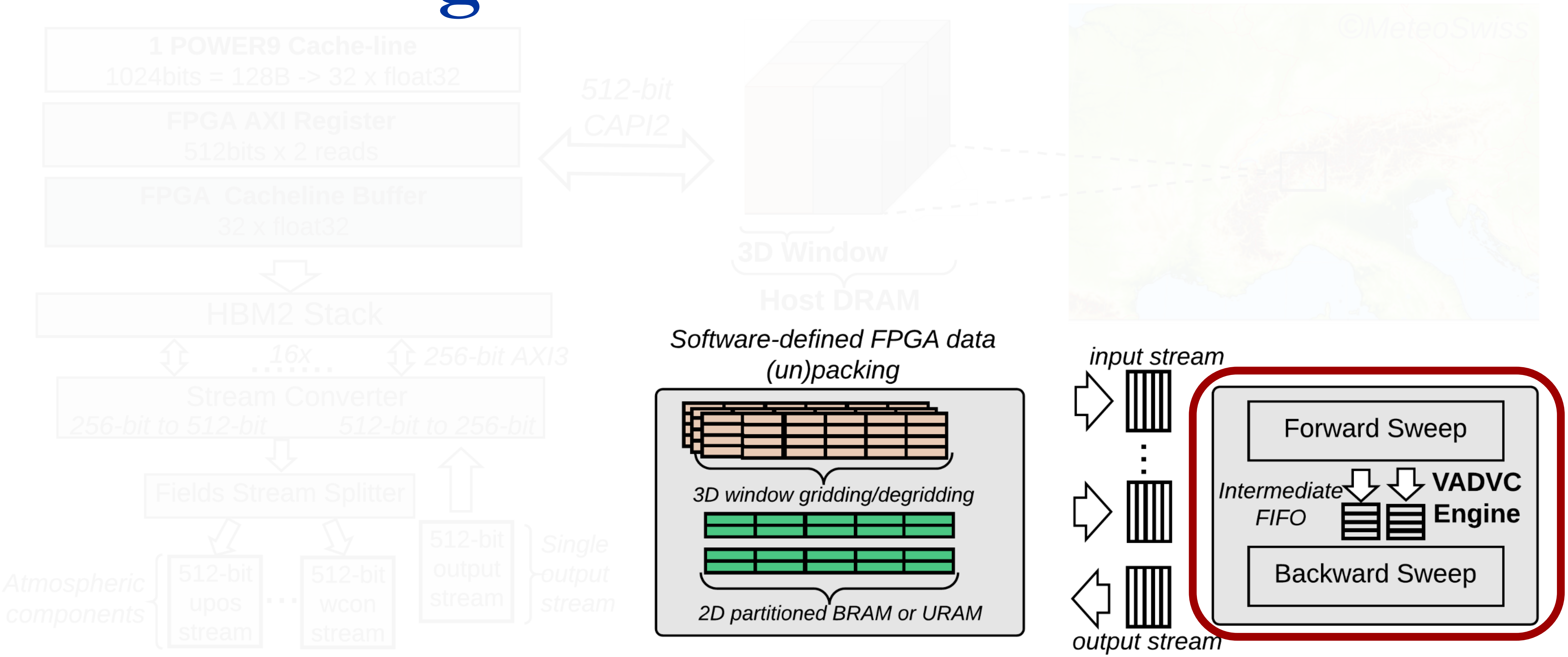
Data mapping onto HBM

NERO Design Flow



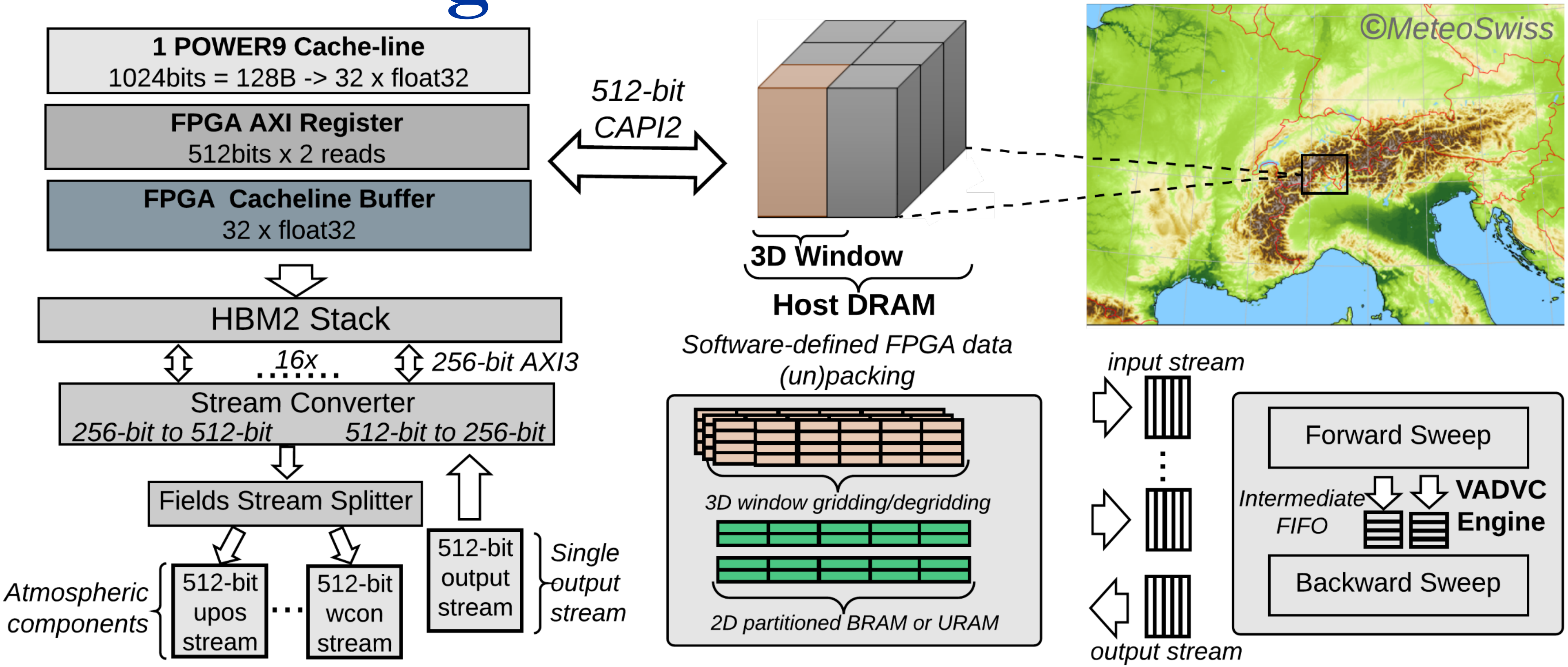
Main execution pipeline

NERO Design Flow



Main execution pipeline

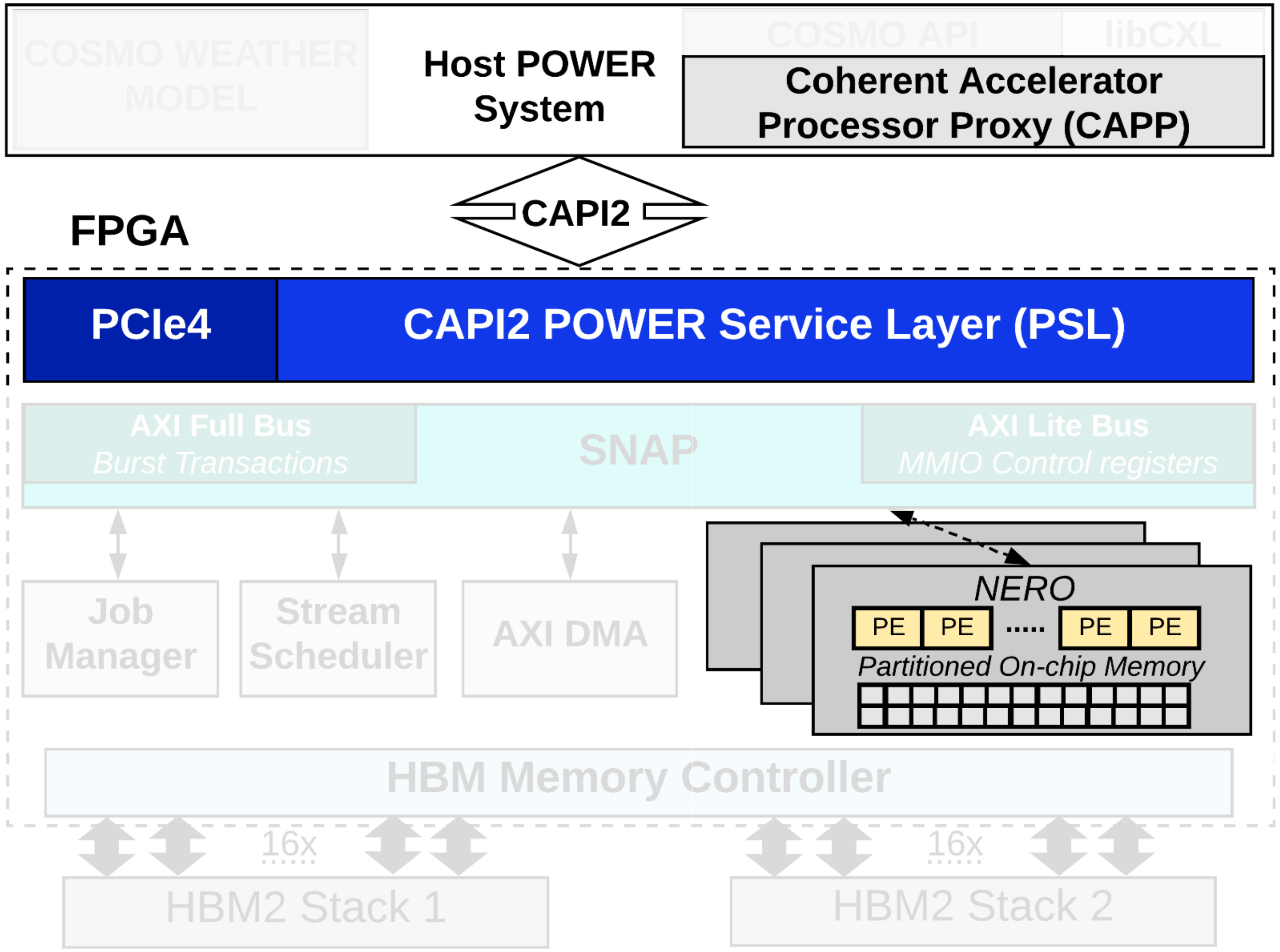
NERO Design Flow



Complete design flow

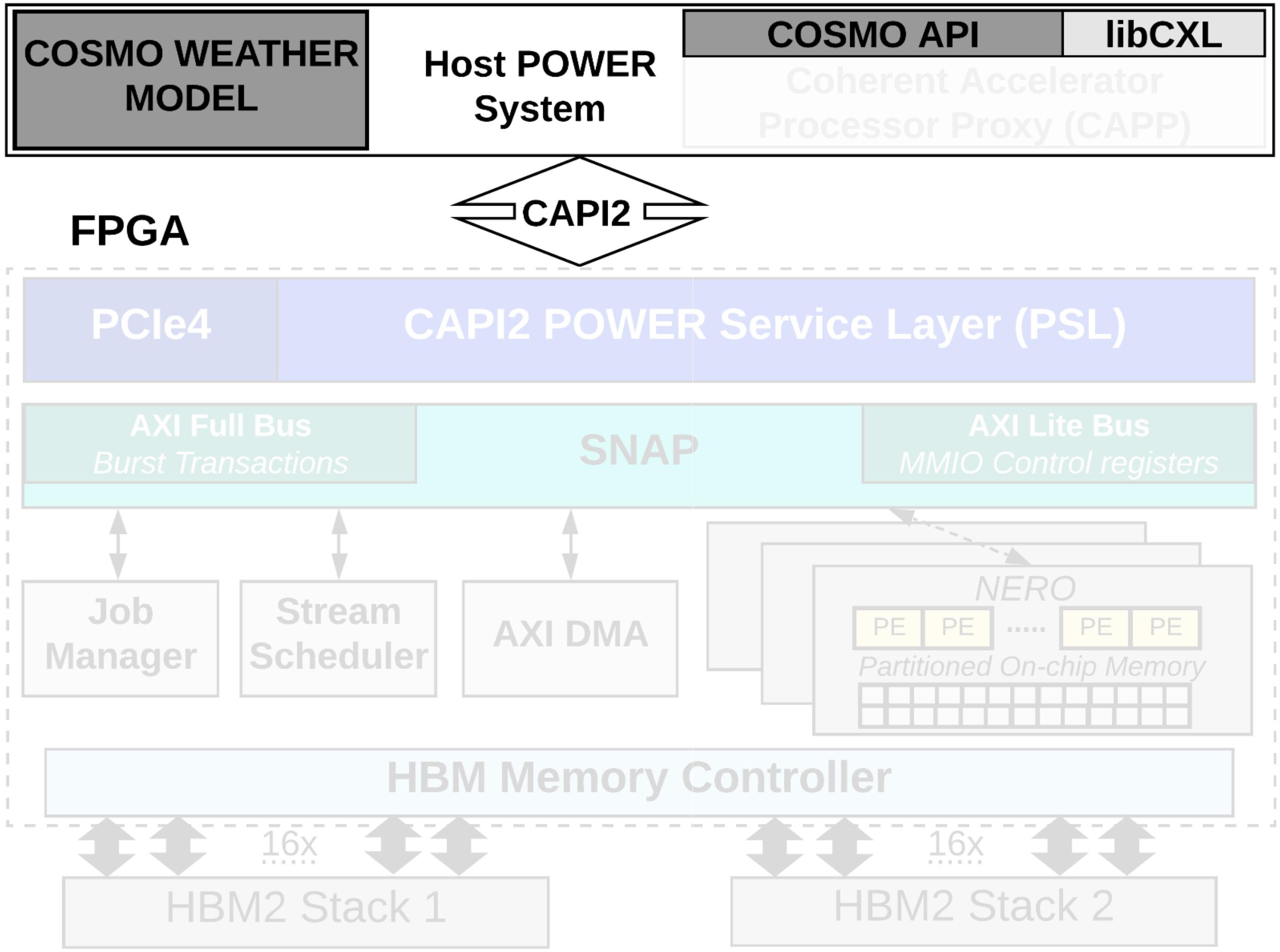
NERO Application Framework

- NERO communicates to Host over **CAPI2** (Coherent Accelerator Processor Interface)



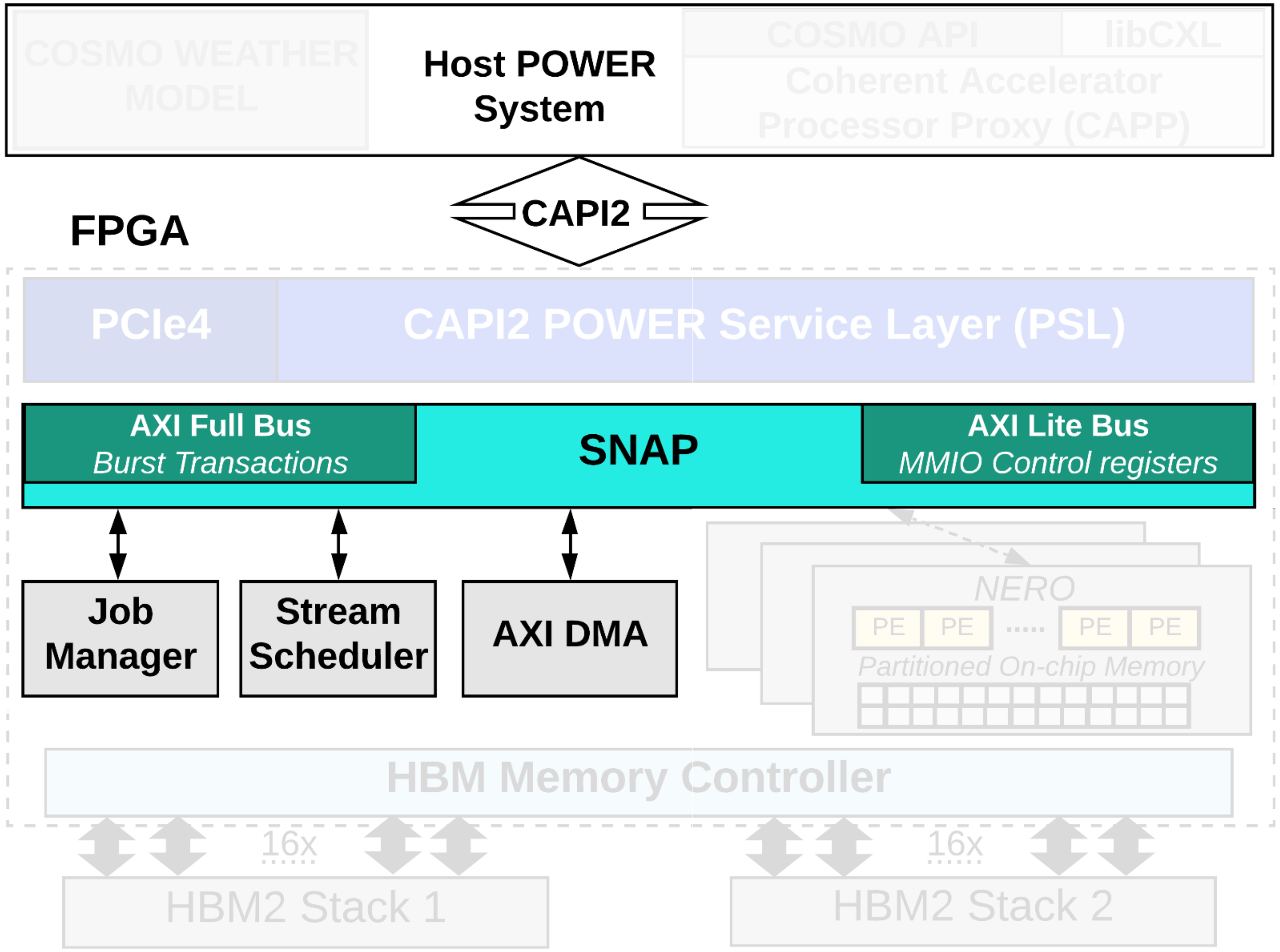
NERO Application Framework

- NERO communicates to Host over **CAPI2** (Coherent Accelerator Processor Interface)
- **COSMO API** handles offloading jobs to NERO



NERO Application Framework

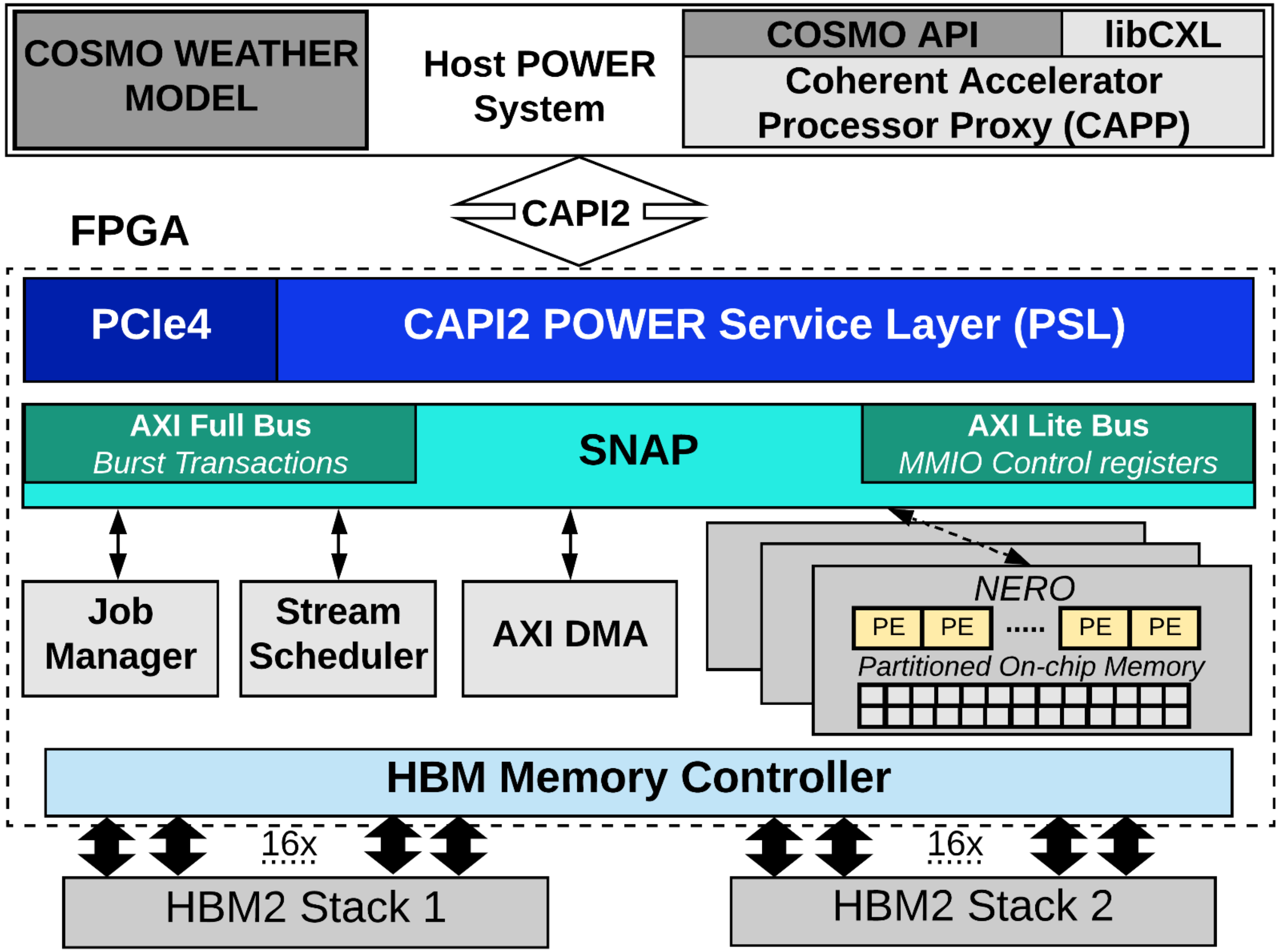
- NERO communicates to Host over **CAPI2** (Coherent Accelerator Processor Interface)
- **COSMO API** handles offloading jobs to NERO
- **SNAP** (Storage, Network, and Analytics Programming) allows for seamless integration of the COSMO API



https://github.com/open-power/snap

NERO Application Framework

- NERO communicates to Host over **CAPI2** (Coherent Accelerator Processor Interface)
- **COSMO API** handles offloading jobs to NERO
- **SNAP** (Storage, Network, and Analytics Programming) allows for seamless integration of the COSMO API



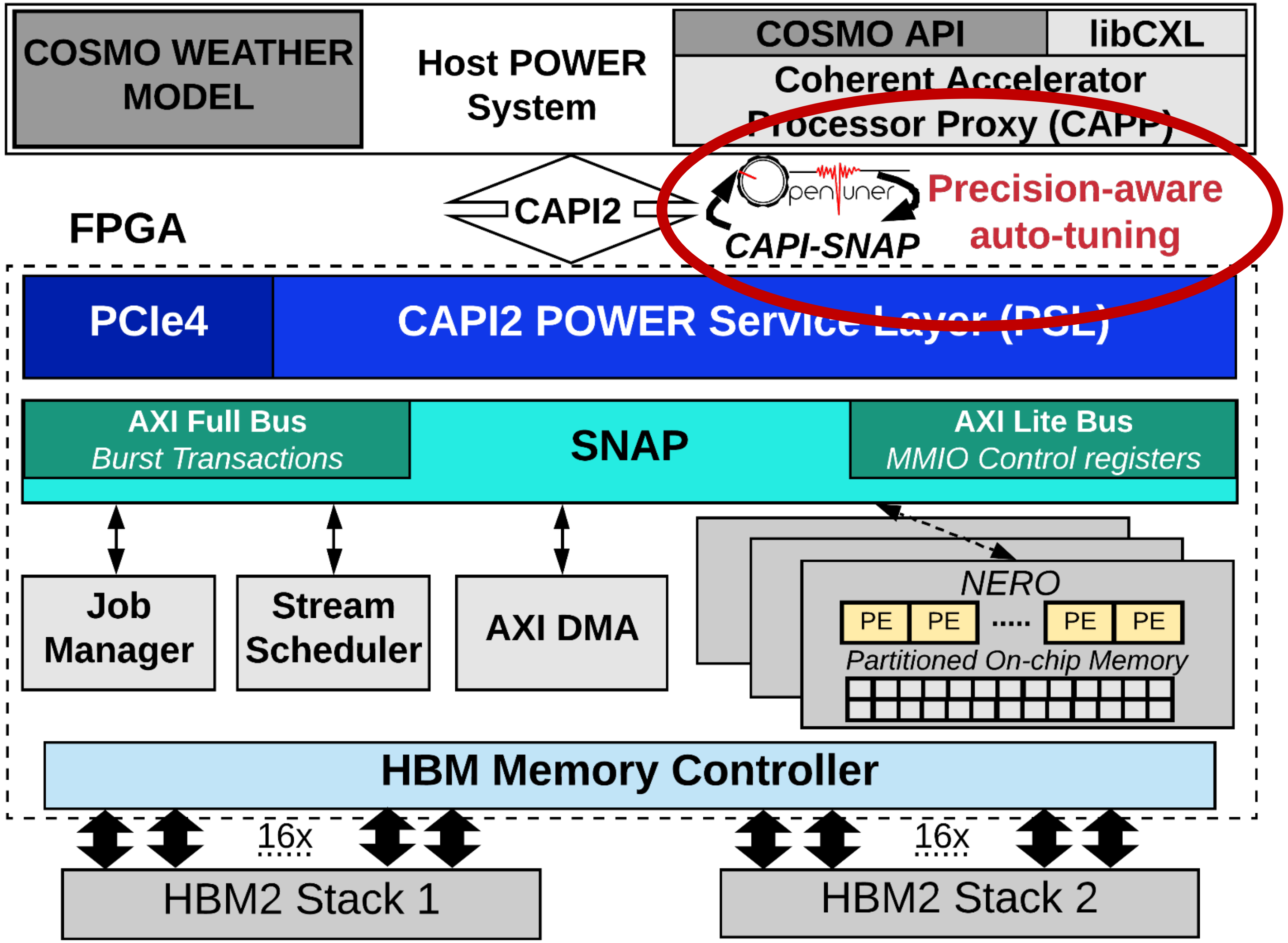
https://github.com/open-power/snap

Outline

- Background
- CPU Roofline Analysis
- FPGA-based Platform
- NERO: **Near-HBM Accelerator for Weather Prediction Modeling**
- Precision-optimized Tiling
- Evaluation
  - Performance Analysis
  - Energy Efficiency Analysis
- Summary

Precision-optimized Tiling

- The **best window size** is **critical**
- Formulate the search for the best window size as a multi-objective **auto-tuning** problem
- Taking into account the **datatype precision**
- We make use of **OpenTuner**

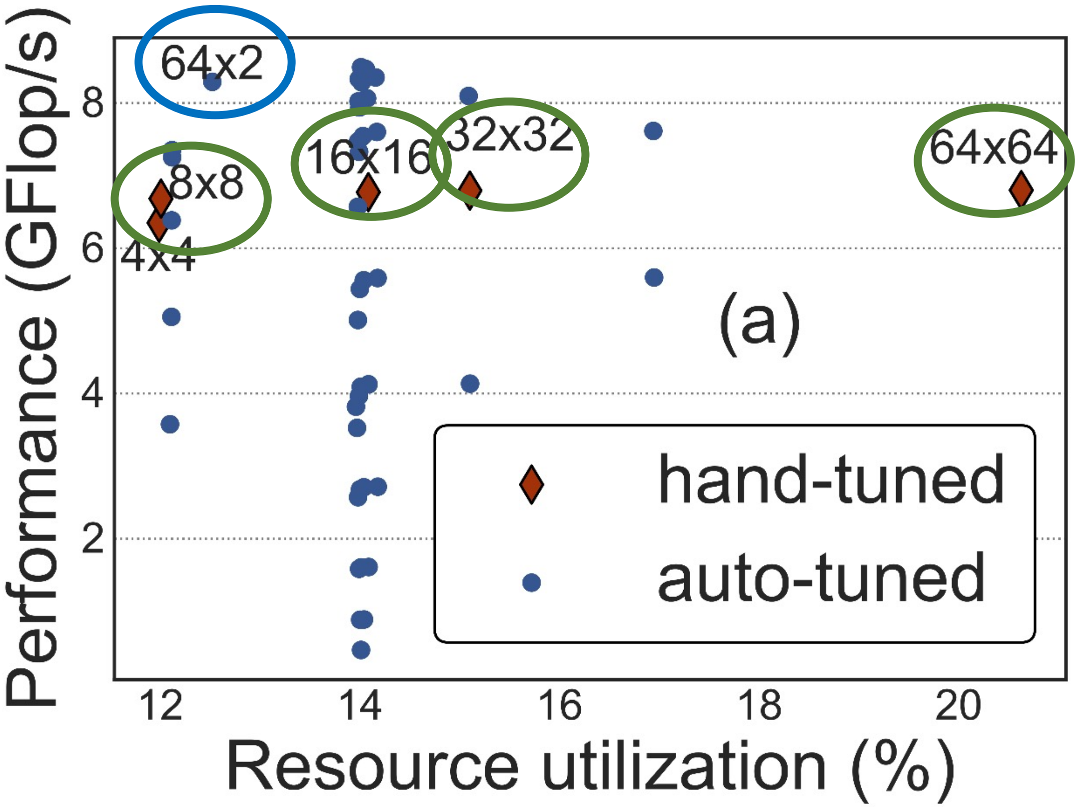


https://github.com/jansel/opentuner

Precision-optimized Tiling

Precision-optimized Tiling

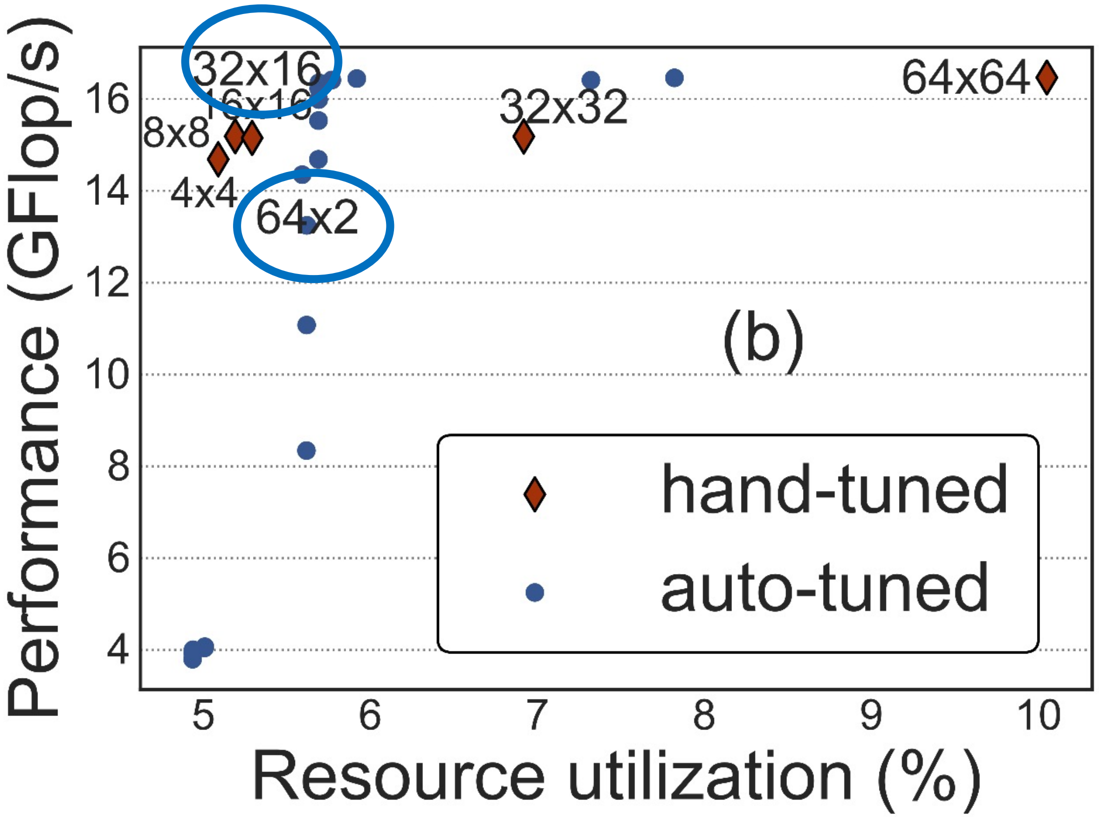
Single Precision



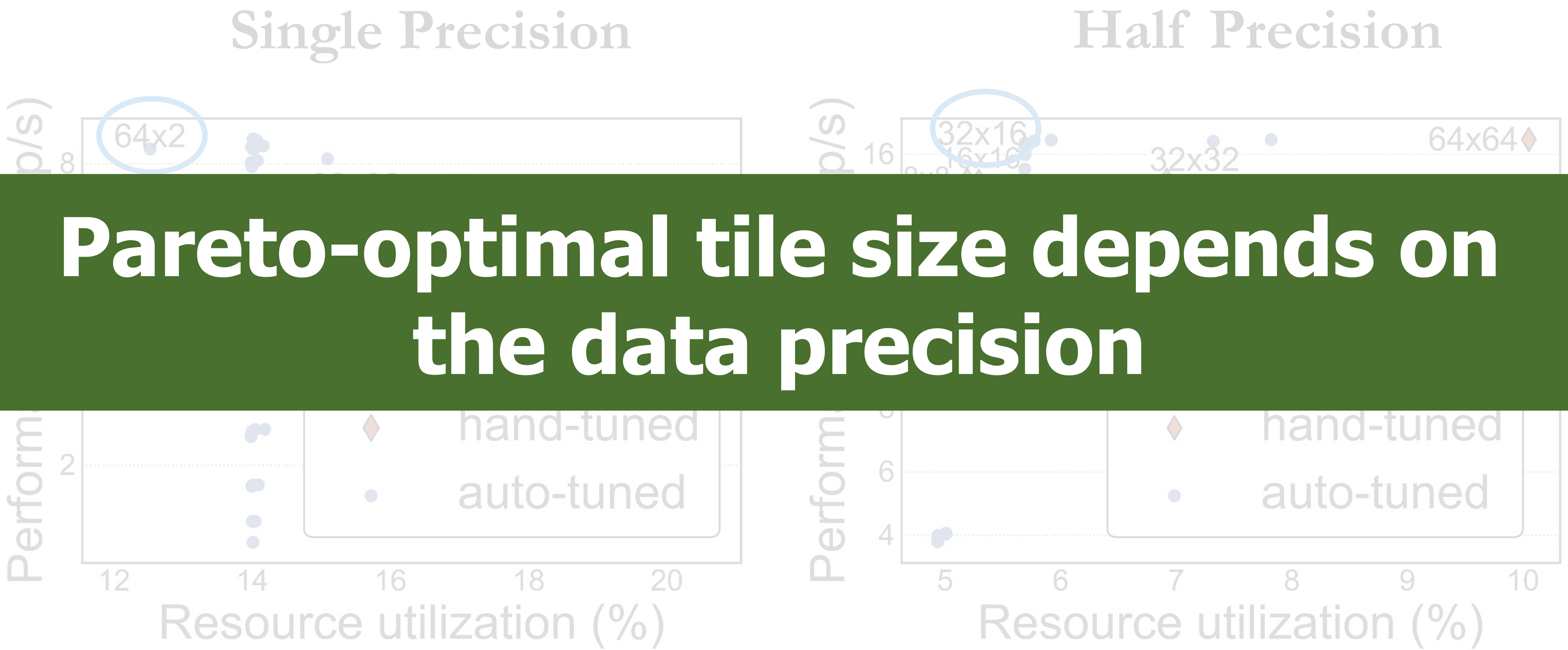
Precision-optimized Tiling

Single Precision

Half Precision



Precision-optimized Tiling



Pareto-optimal tile size depends on the data precision

Outline

Background

CPU Roofline Analysis

FPGA-based Platform

NERO: **Near-HBM Accelerator for Weather Prediction Modeling**

Precision-optimized Tiling

Evaluation

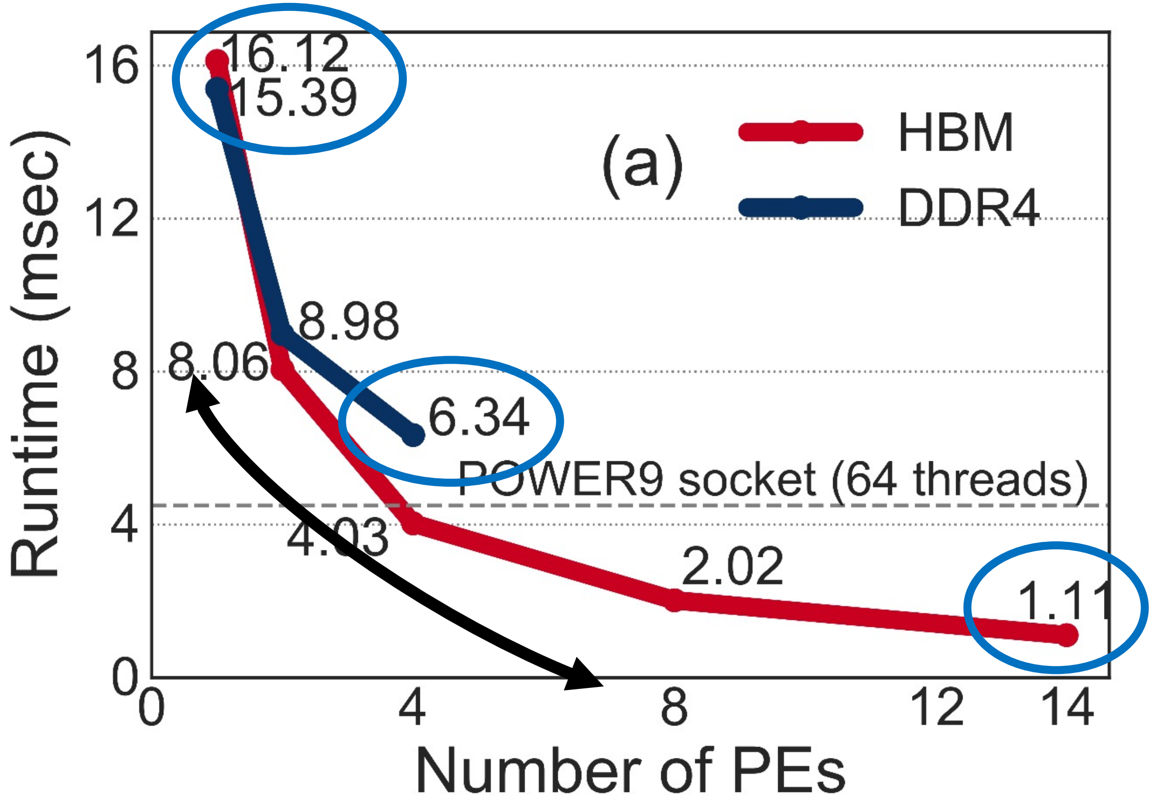
Performance Analysis

Energy Efficiency Analysis

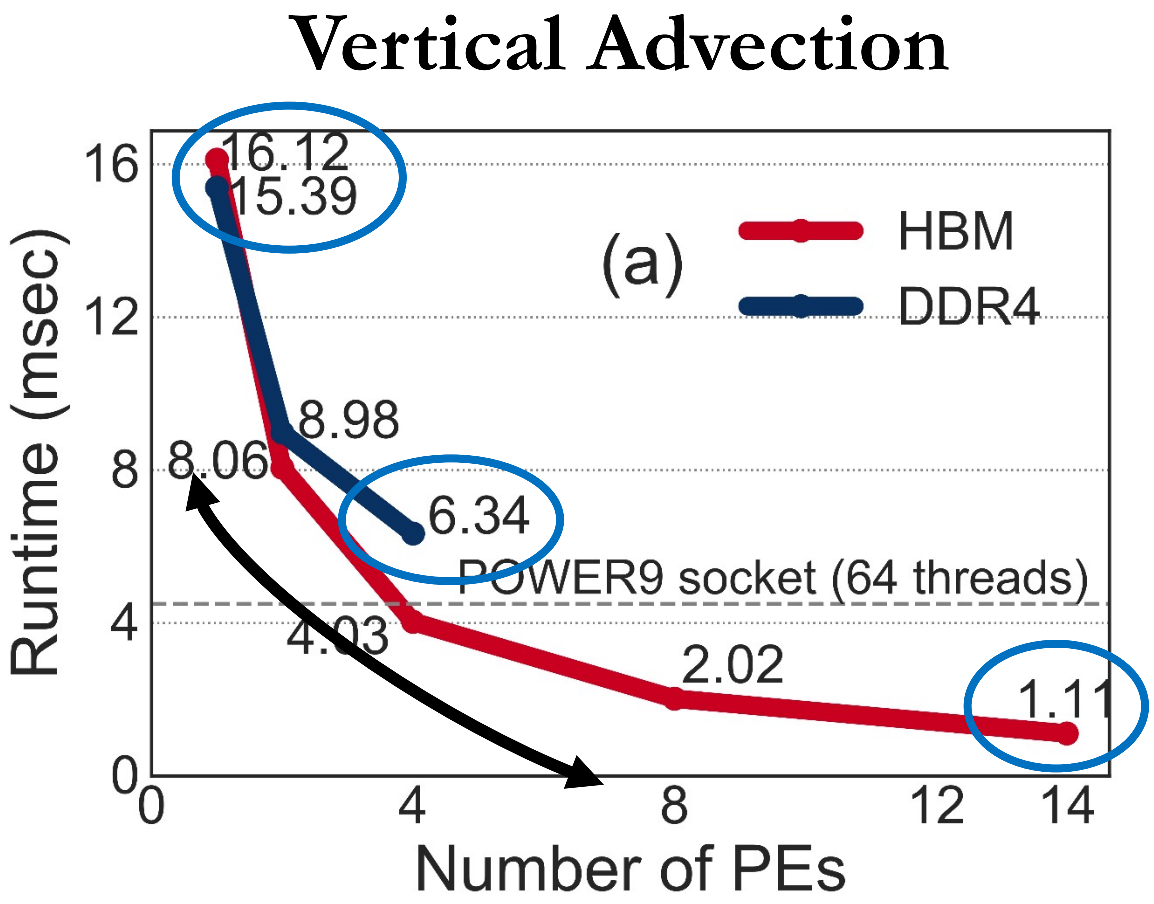
Summary

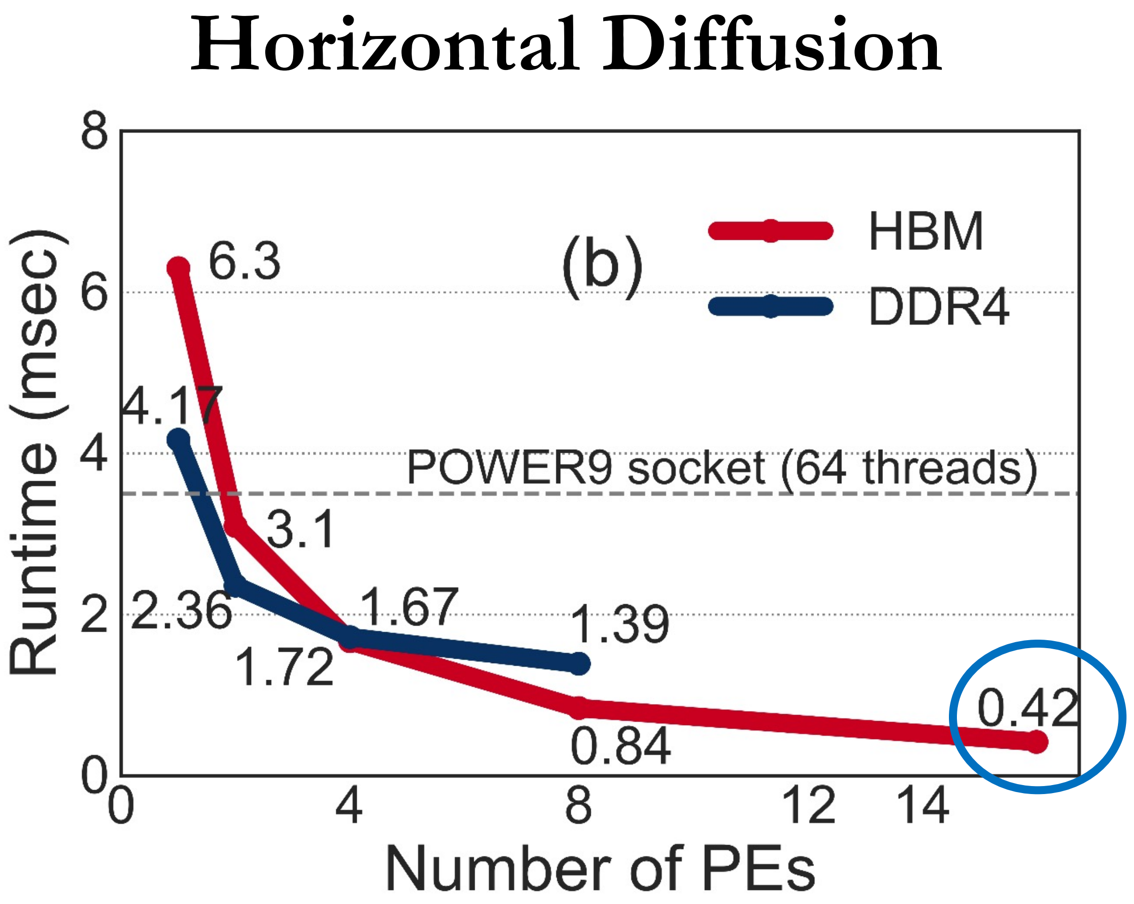
NERO Performance Analysis

Vertical Advection

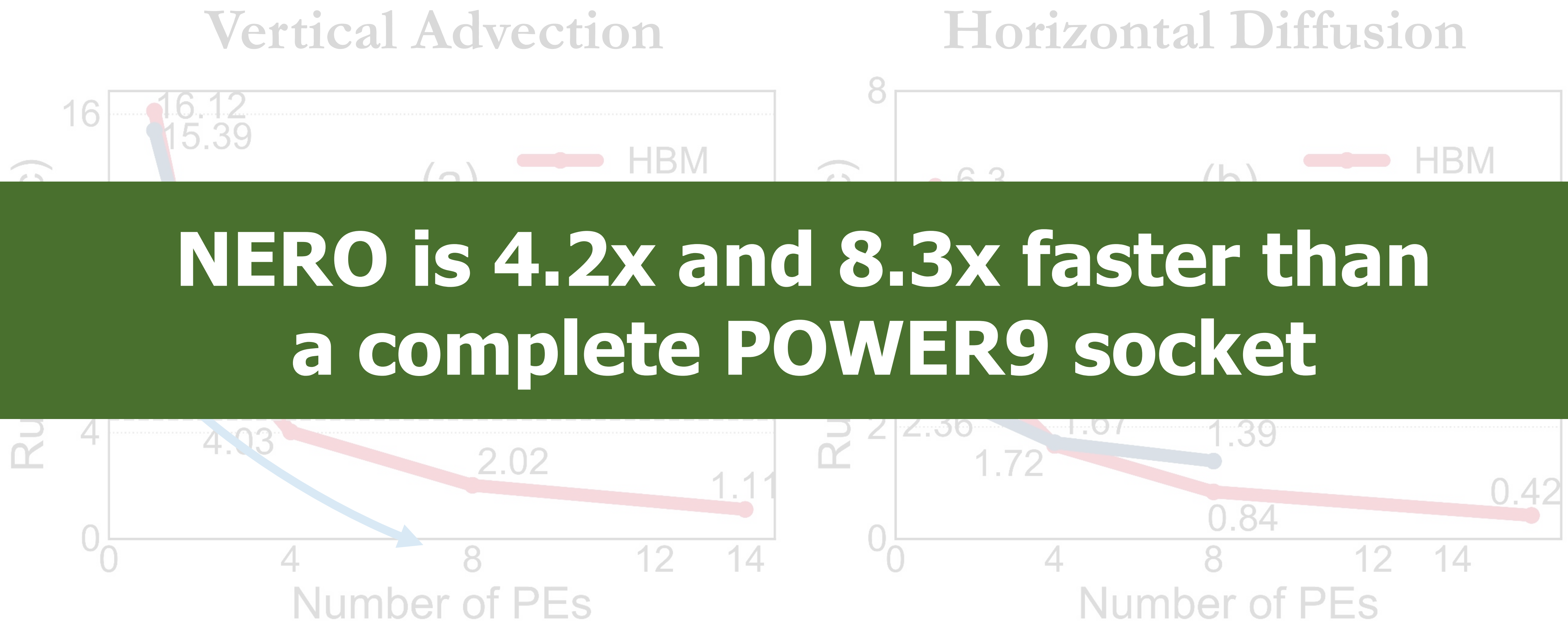


NERO Performance Analysis





NERO Performance Analysis

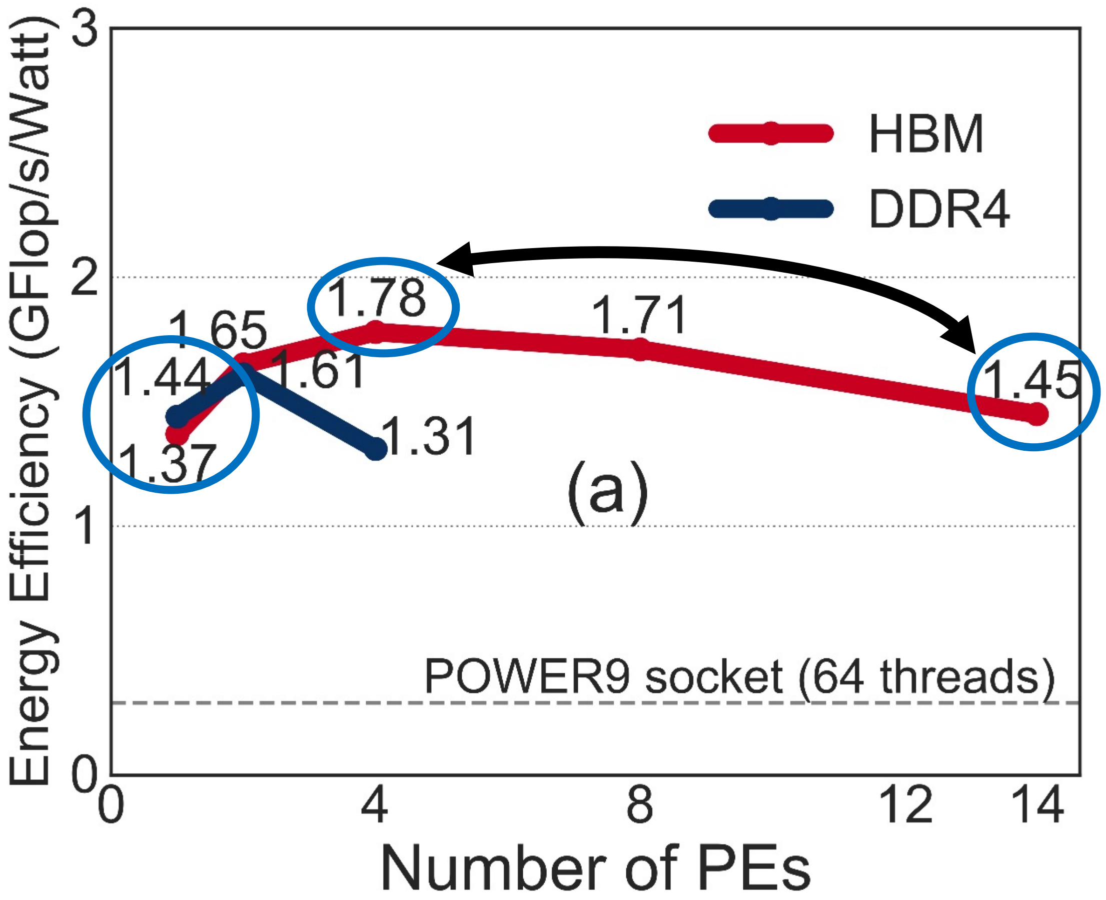


NERO is 4.2x and 8.3x faster than a complete POWER9 socket

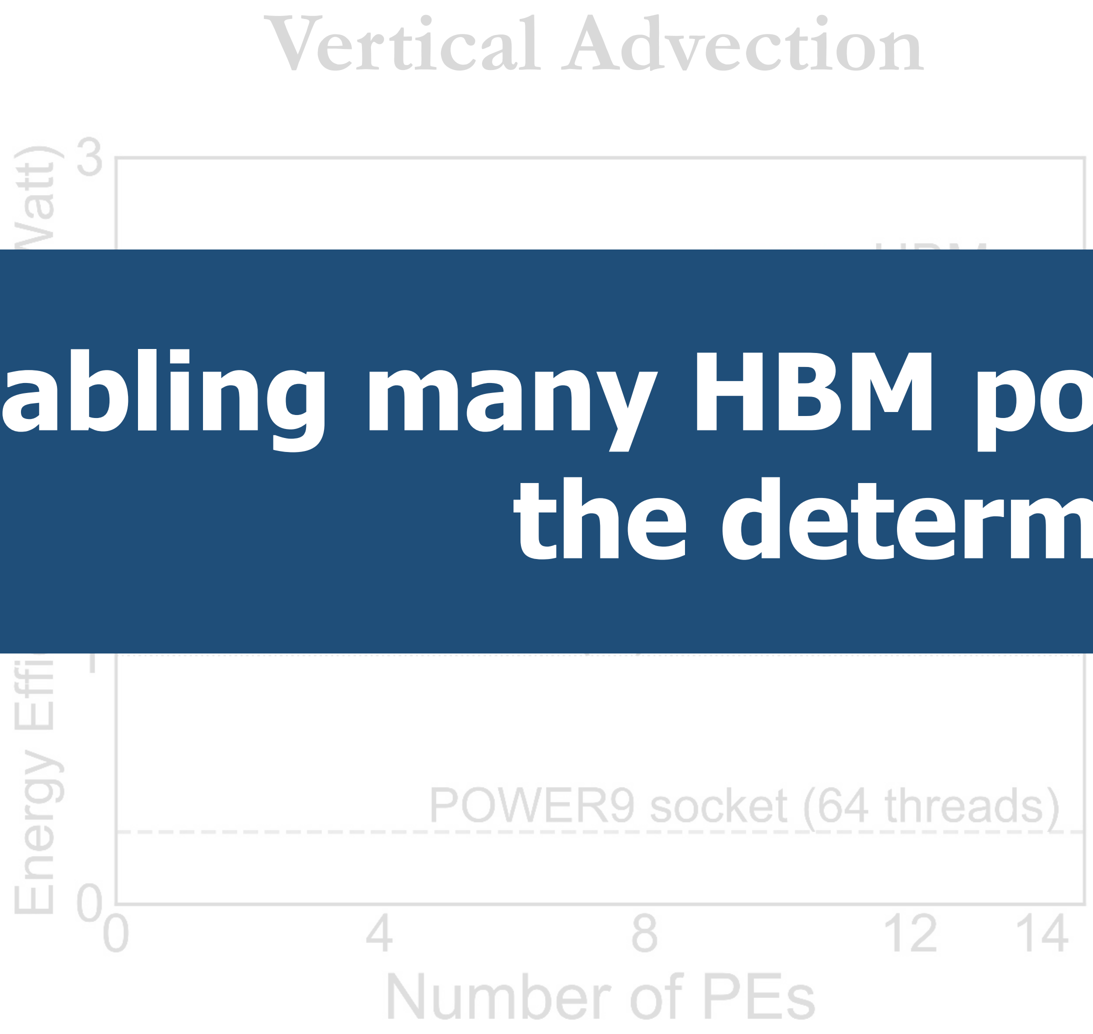
Outline

How Energy Efficient is NERO?

Vertical Advection

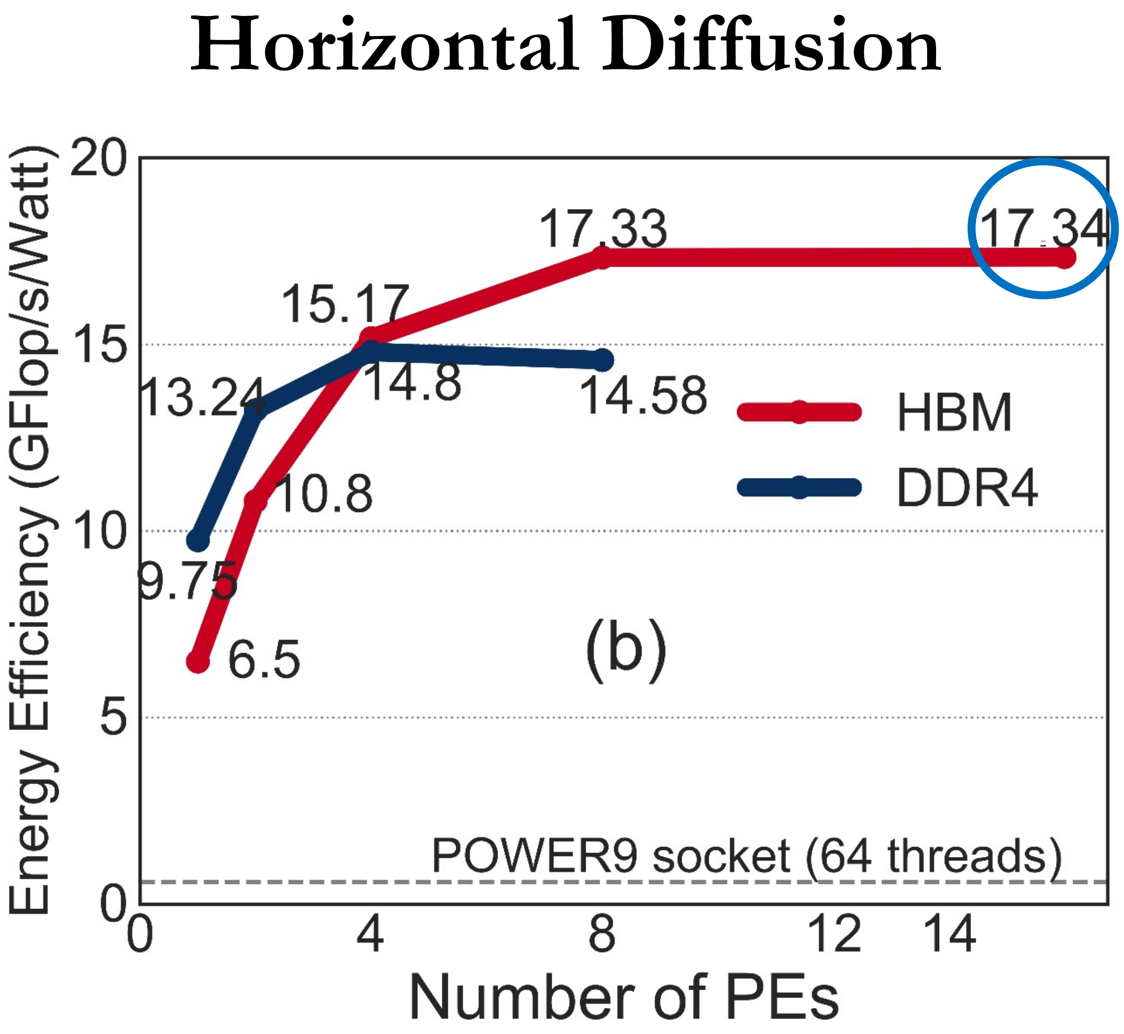


How Energy Efficient is NERO?



Enabling many HBM ports might not always be the determining factor

How Energy Efficient is NERO?



How Energy Efficient is NERO?

NERO reduces energy consumption by 22x and 29x compared to a complete POWER9 socket

NERO provides energy efficiency of 1.5 GFLOPS/Watt and 17.3 GFLOPS/Watt

Outline

Background

CPU Roofline Analysis

FPGA-based Platform

NERO: **Near-HBM Accelerator for Weather Prediction Modeling**

Precision-optimized Tiling

Evaluation

- Performance Analysis
- Energy Efficiency Analysis

Summary

Summary

- **Motivation:** Stencil computation is an essential part of weather prediction applications
- **Problem:** Memory bound with limited performance and high energy consumption on multi-core architectures
- **Goal:** Mitigate the performance bottleneck of compound weather prediction kernels in an energy-efficient way
- **Our contribution: NERO**
  - First near High-Bandwidth Memory (HBM) FPGA-based accelerator for representative kernels from a real-world weather prediction application
  - Detailed roofline analysis to show weather prediction kernels are constrained by DRAM bandwidth on a state-of-the-art CPU system
  - Data-centric caching with precision-optimized tiling for a heterogeneous memory hierarchy
  - Scalability analysis for both DDR4 and HBM-based FPGA boards
- **Evaluation**
  - NERO outperforms a 16-core IBM POWER9 system by 4.2x and 8.3x when running two compound stencil kernels
  - NERO reduces energy consumption by 22x and 29x with an energy efficiency of 1.5 GFLOPS/Watt and 17.3 GFLOPS/Watt

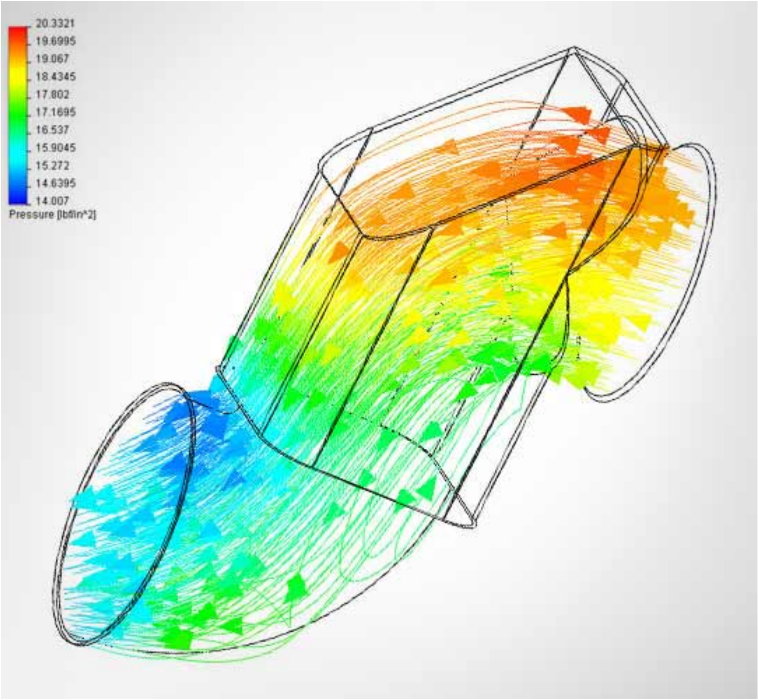
Low Precision Processing for High Order Stencil Computation

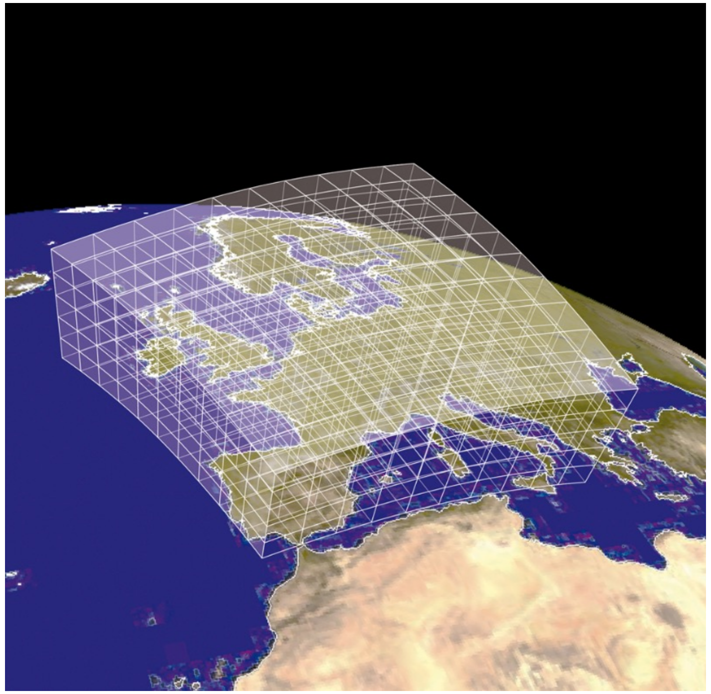
Executive Summary

- **Motivation:** Low precision computing is a promising approach to **solve data movement bottleneck** for emerging **big data workloads**
- **Problem:** A key barrier to a widespread adoption of reduced-precision computing is the lack of an architecture exploiting arbitrary precision, supported by a software layer that controls the precision of computations
- **Our contribution:**
  - Systematic precision exploration for various 3D stencils for a wide range of number systems-fixed, float, posit
  - Using a state-of-the-art multi-core CPU with FPGA to show the capability of reduced precision
- **Evaluation**
  - **50% lower bits with only 1% loss of accuracy for all the number systems**
  - **Lower precision leads to ~FPGA peak performance of 468-659 GOP/s with 30-50x higher energy efficiency**

Stencil Computations and Applications

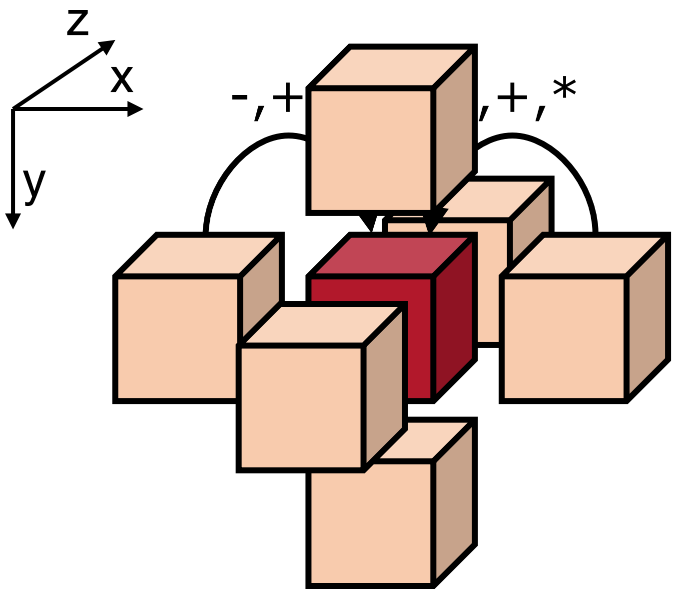
- Stencils are widely used in many applications:
  - fluid dynamics, image processing, atmospheric modelling

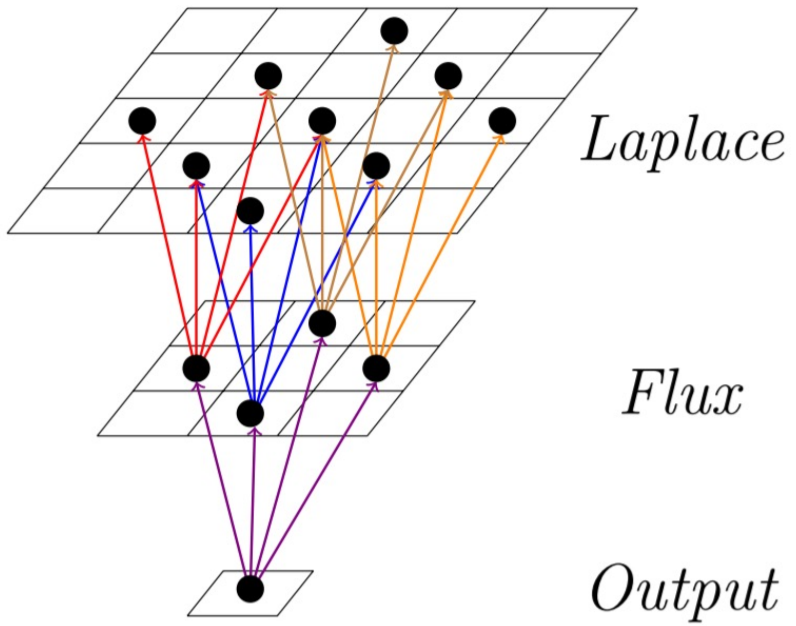




Application Structure

- Stencil is computed using some elementary operations (e.g. weighted difference)
- Stencil operates on high-order (multi-dimensional) field/array
- Often consists of multiple update step





Workload characteristics

- High-order stencil computations are cache unfriendly
  - Limited arithmetic intensity: only reuse potential in neighboring pixels
  - Sparse and complex access pattern:

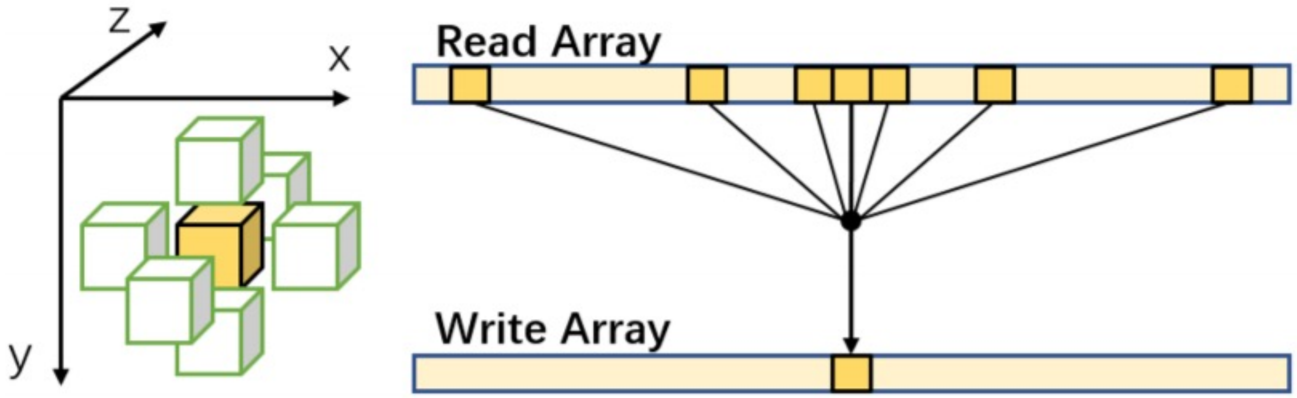
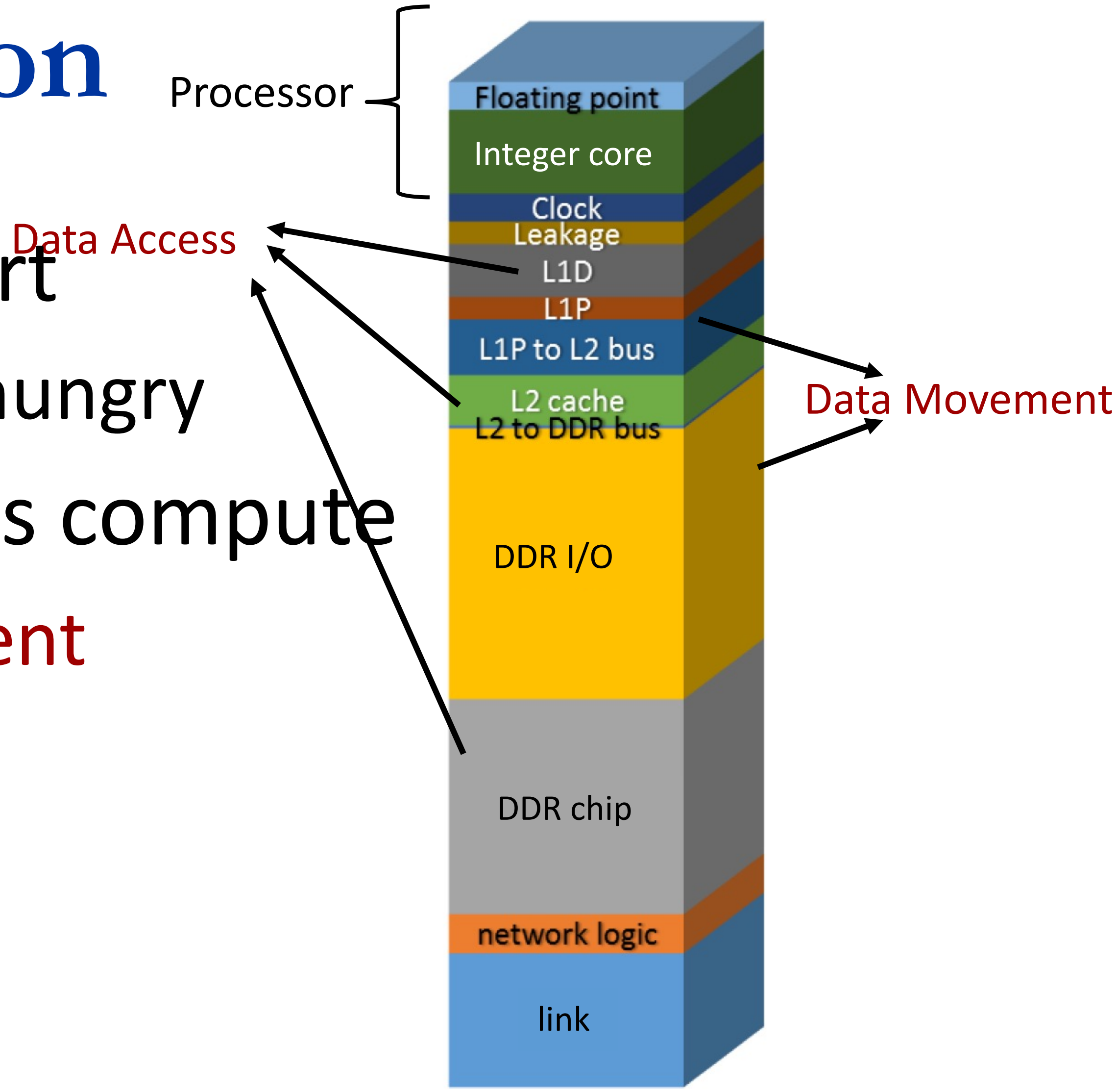


Image source: Performance Tuning and Analysis for Stencil-Based Applications on POWER8 Processor – Xu et al. (2018)

Conventional Computation

- Data access consumes a major part
  - Applications are increasingly data hungry
- Data movement energy dominates compute
  - Especially true for off-chip movement

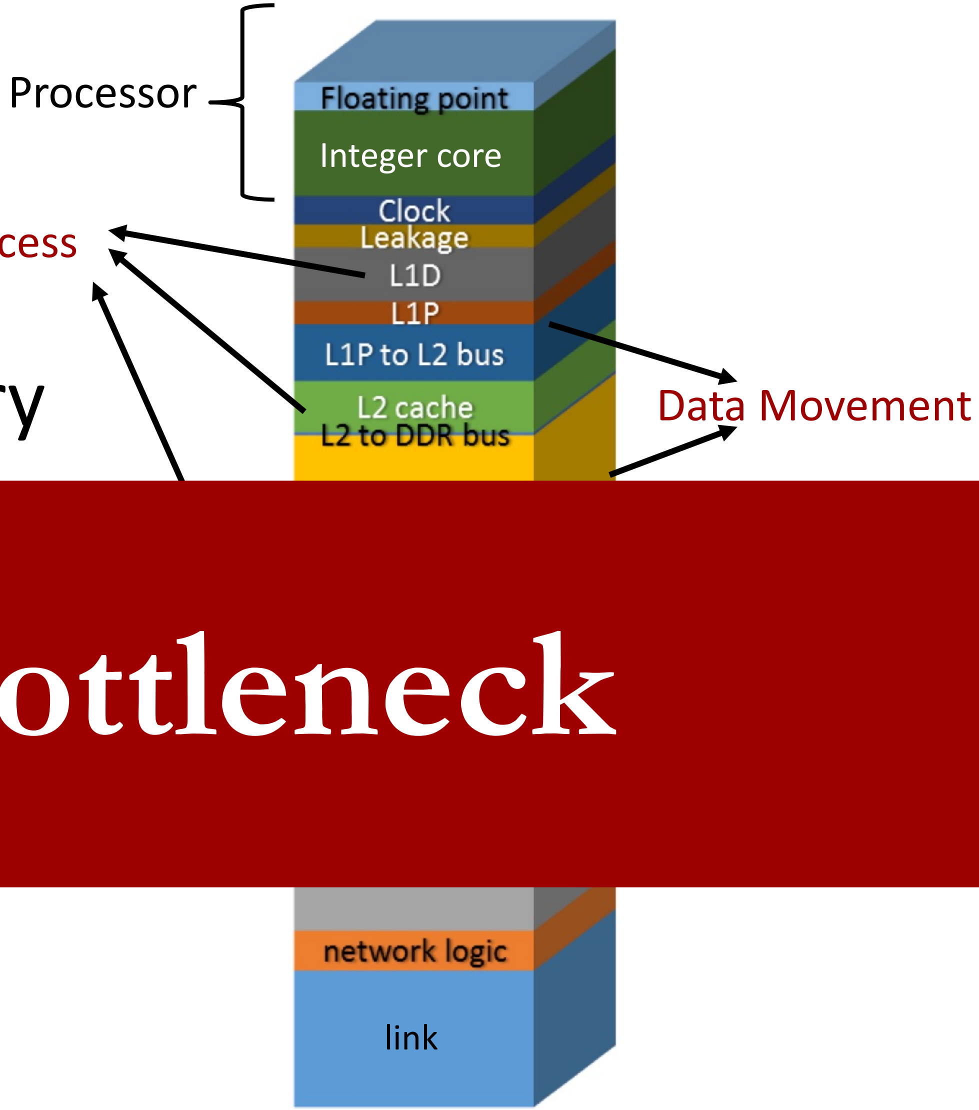


System-level power break down*

** R. Nair et al., "Active memory cube: A processing-in memory architecture for exascale systems", IBM J. Research Develop., vol. 59, no. 2/3, 2015*

Conventional Computation

- Data access consumes a major part
  - Applications are increasingly data hungry



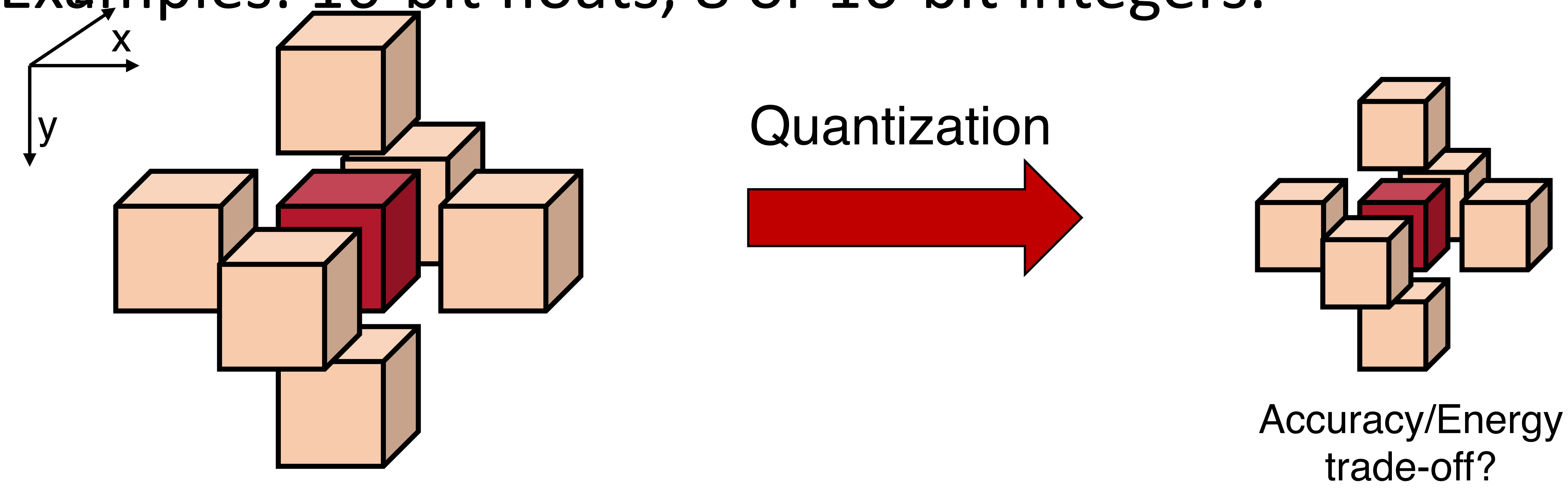
System-level power break down*

Data movement bottleneck

** R. Nair et al., "Active memory cube: A processing-in memory architecture for exascale systems", IBM J. Research Develop., vol. 59, no. 2/3, 2015*

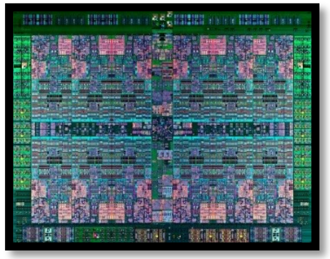
Reduced-Precision Computations

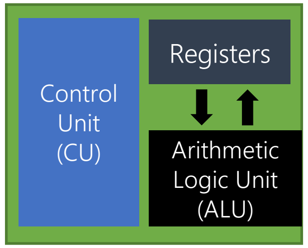
- Stencil computations generally use a high-precision number format
- Many emerging applications use reduced-precision data types
  - Examples: 16-bit floats, 8 or 16-bit integers.



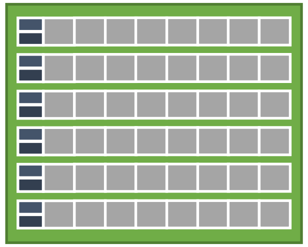
Alternative platforms

CPUs

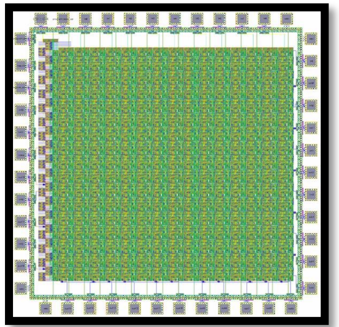


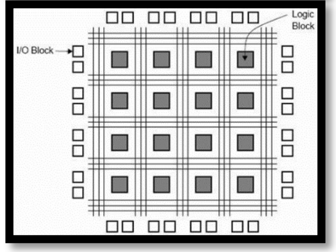


GPUs

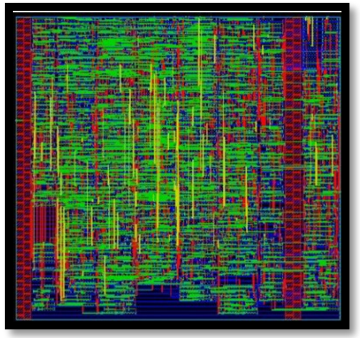


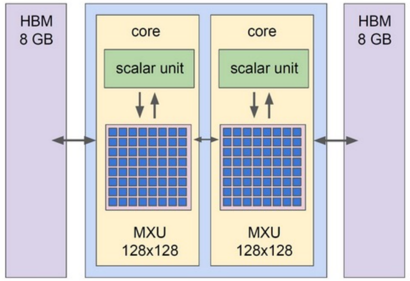
FPGAs





ASICs





Processing Units ASICs for emerging workloads, e.g. Google TPU

FLEXIBILITY

EFFICIENCY

FPGAs ideal for adapting to rapidly evolving workloads!

Problem statement

- Stencils have many applications, but difficult to map to traditional platforms
- Low precision computing is a promising approach to solve data movement bottleneck for emerging big data workloads
- FPGAs might enable energy-efficient mapping of various stencil applications

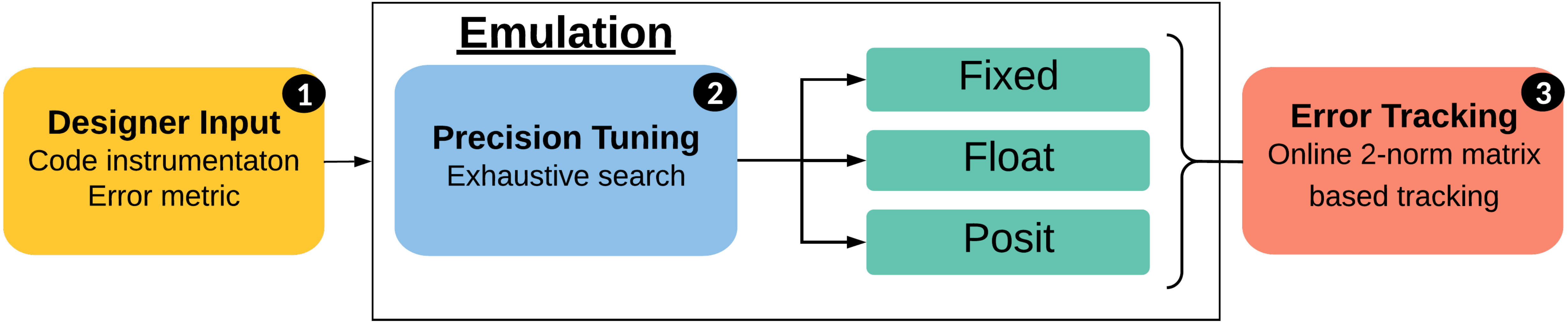
Main contributions:

- Systematic exploration of reduced-precision number formats for stencils
- A case study on a state-of-the-art IBM MPSoC + FPGA platform

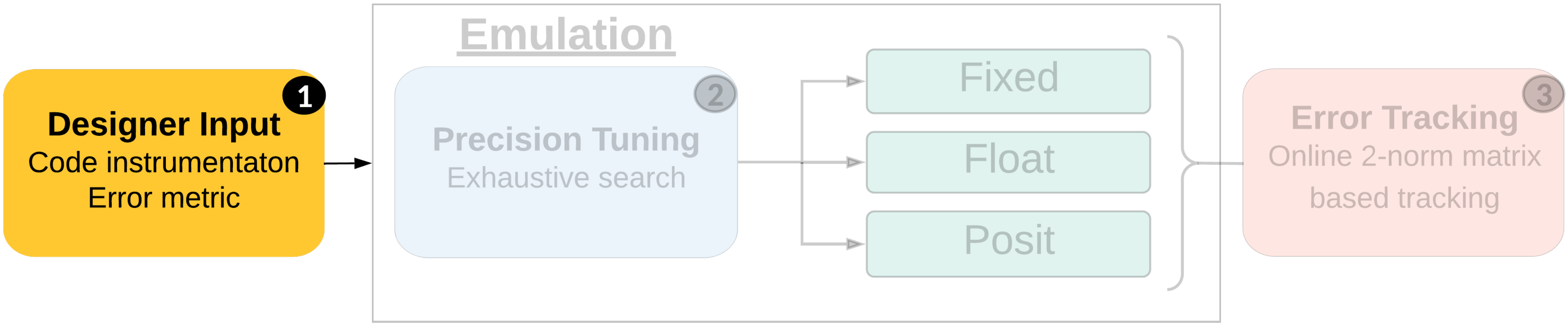
Outline

- Introduction
- **Precision exploration**
- Evaluation on MPSoC + FPGA platform
- Conclusions

Precision Exploration Methodology

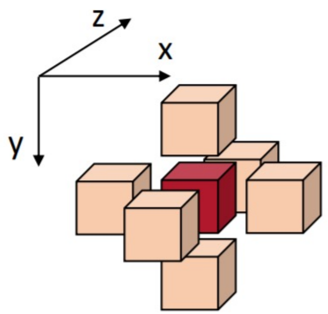


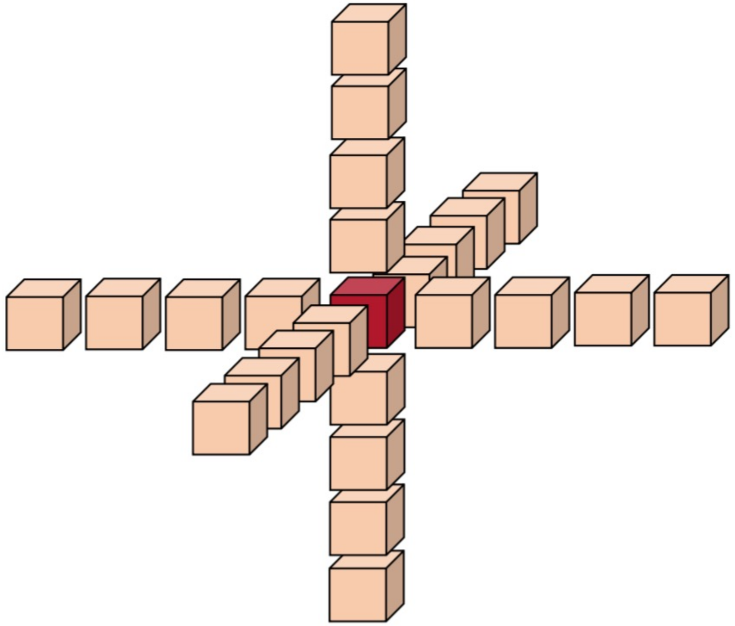
Step 1: Code Instrumentation

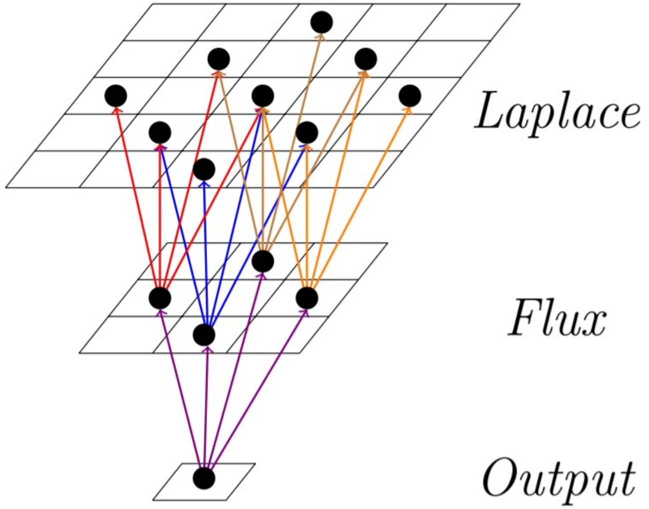


High-order stencil benchmarks

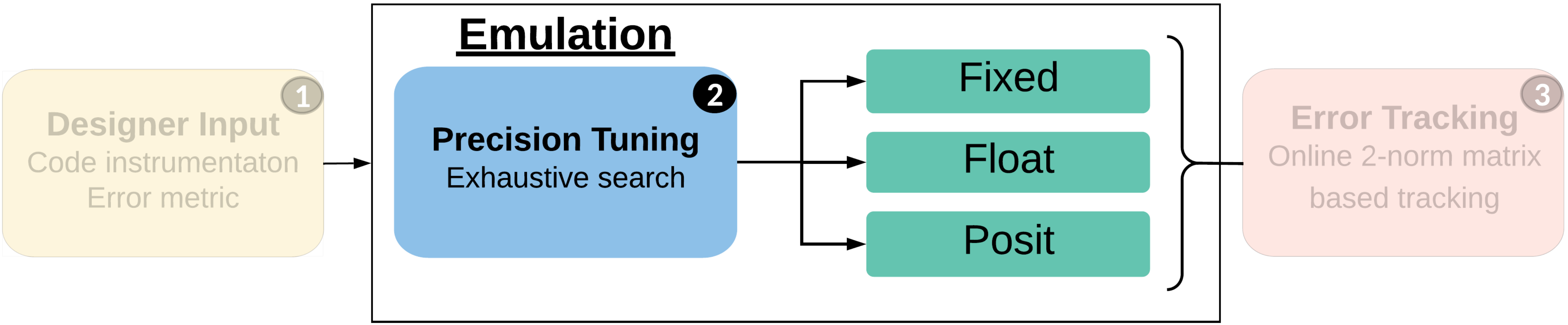
- Elementary stencil: 7 and 25 points
- Compound stencil: horizontal diffusion
- Sweep over a 3D grid with 1280 x 1080 x 960 output pixels





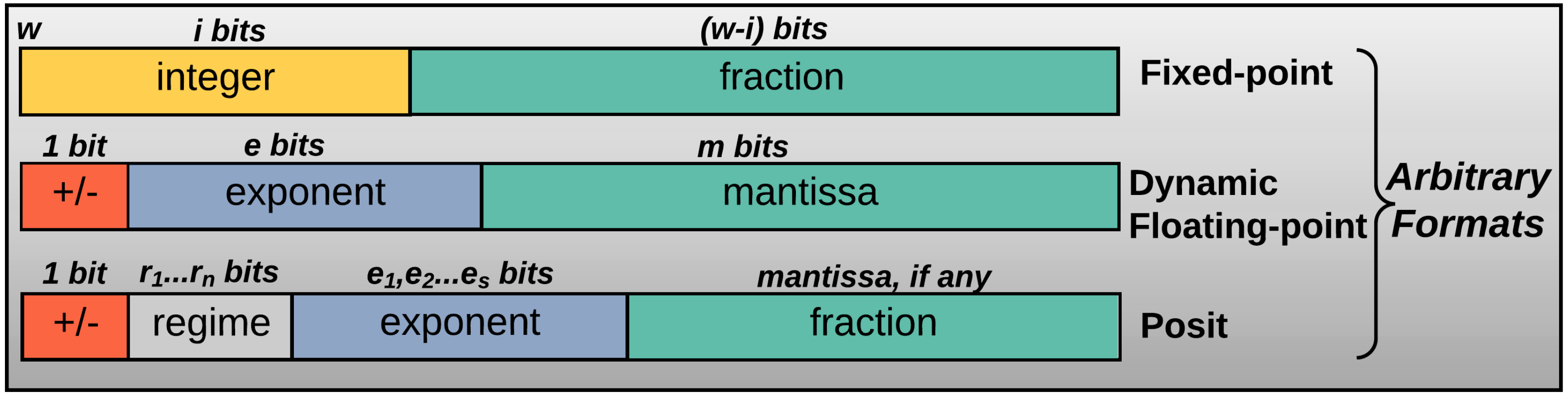


Step 2: Precision Tuning



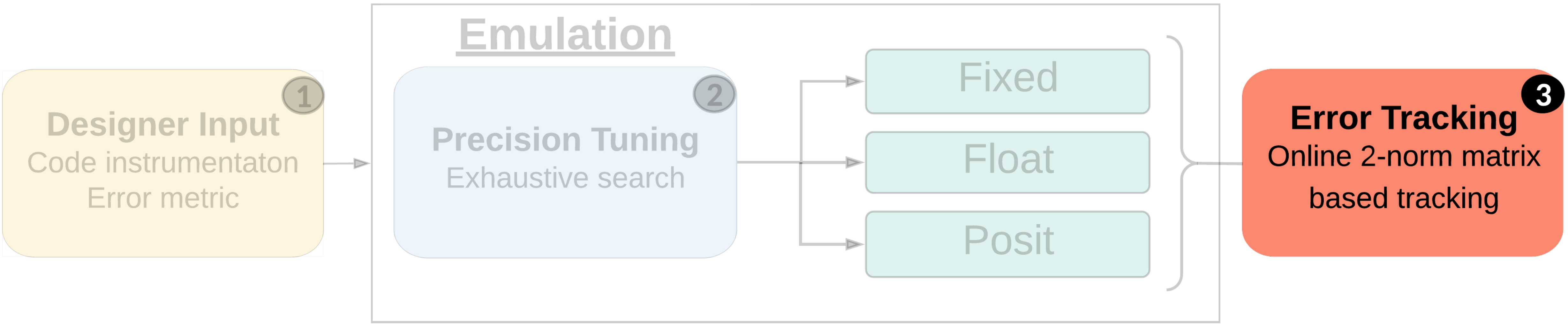
Arbitrary Number Formats

- Fixed-point- *Xilinx fixed-point library from the Vivado 2018.2*
- Dynamic Floating-point –*Floatx library* [1]
- Posit- *Universal number system* [2]



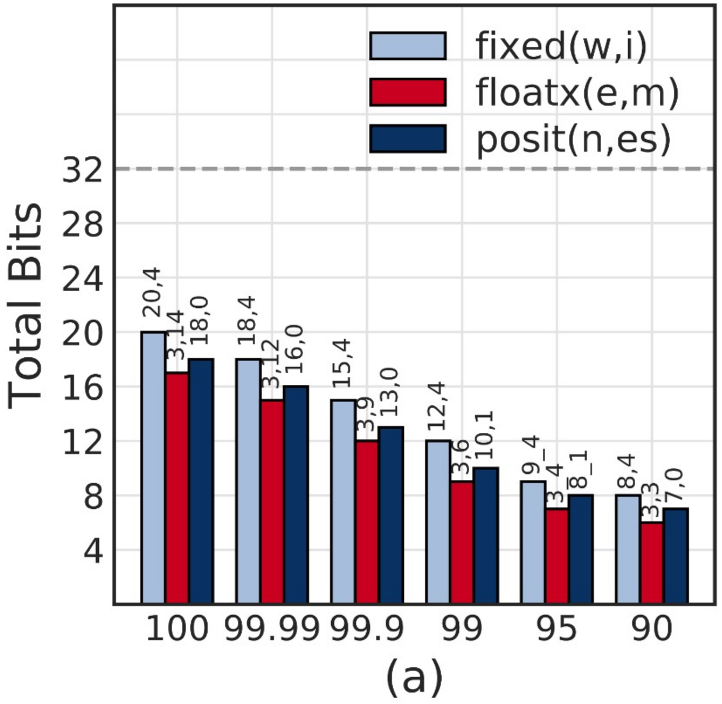
[1]https://github.com/oprecomp/FloatX
[2]https://github.com/stillwater-sc/universal

Step 3: Error Tracking



Results – Emulated Precision Tuning

- Float and Posit obtain full accuracy with less bits
- Significant bit width reduction with accuracy loss of 1%
- Compound stencils require higher dynamic range than 7 and 25 kernel



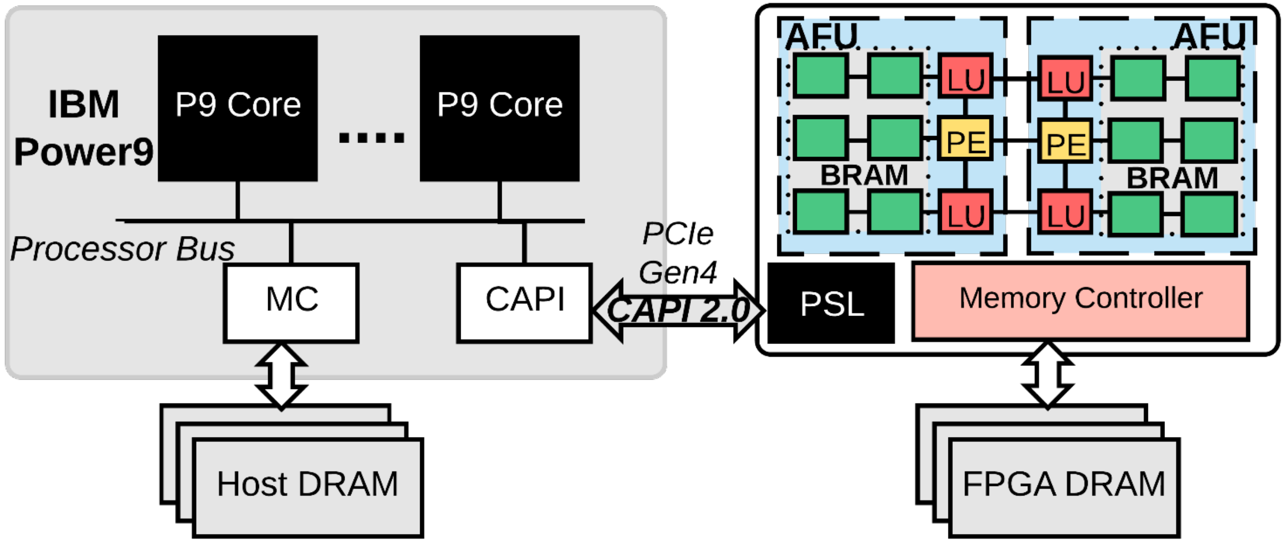
7-point

Outline

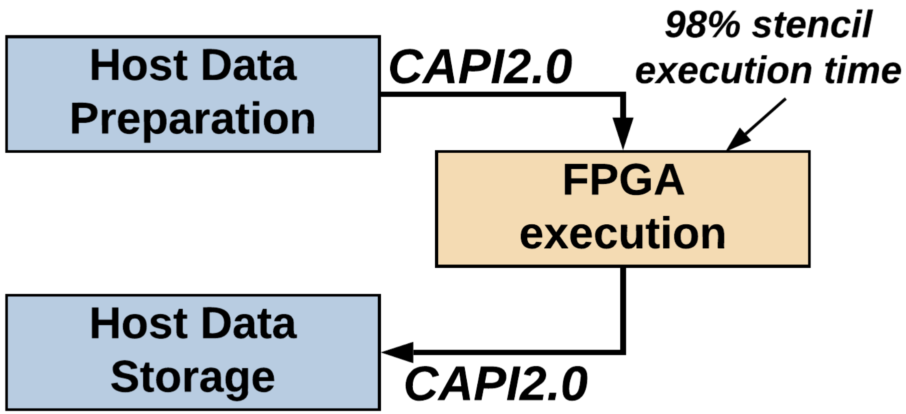
- Introduction
- Precision exploration
- **Evaluation on MPSoC + FPGA platform**
- Conclusions

Case Study: CPU+FPGA

- Host System
  - IBM POWER9
- FPGA board
  - Xilinx Virtex® Ultrascale+™ XCVU3P-2
- Power: IBM AMESTER[3]

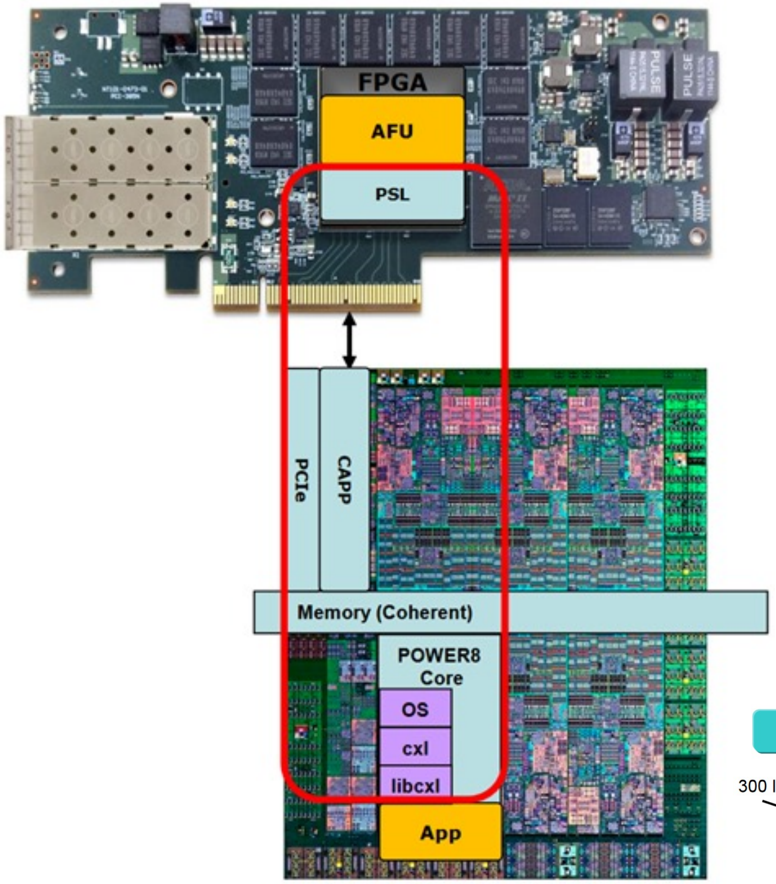


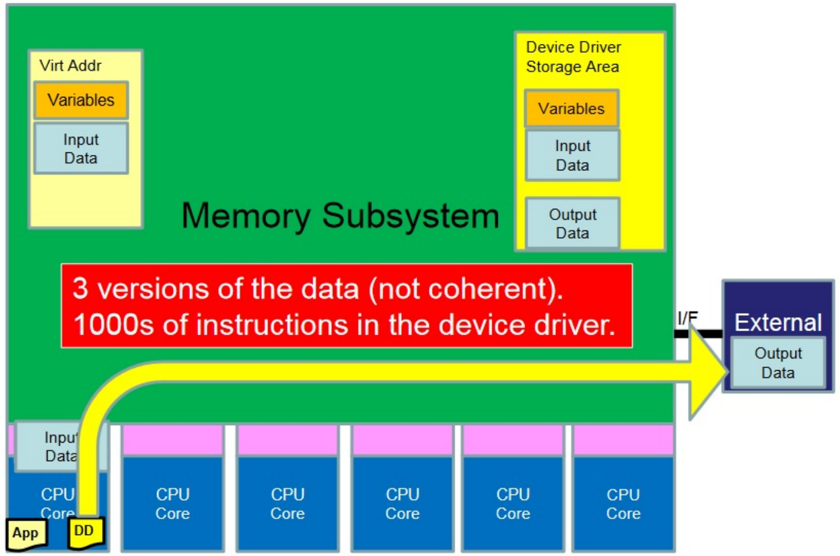
CPU-FPGA co-design execution flow

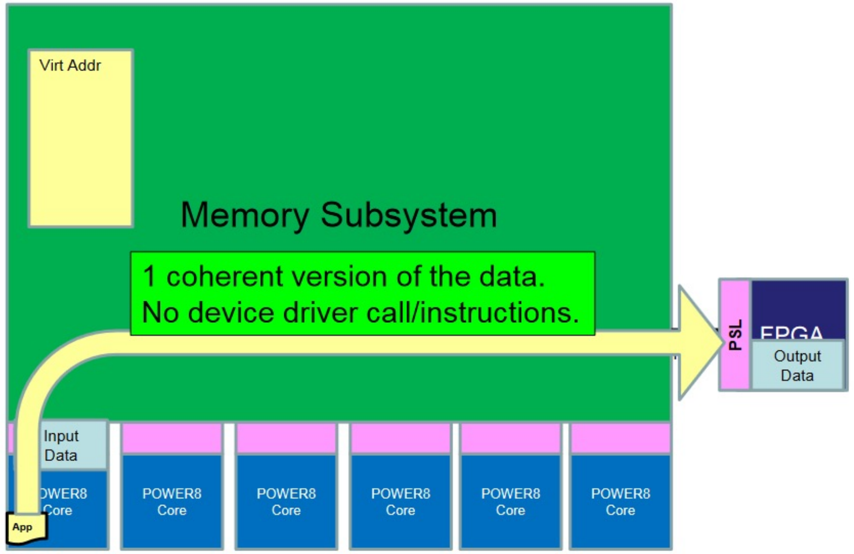


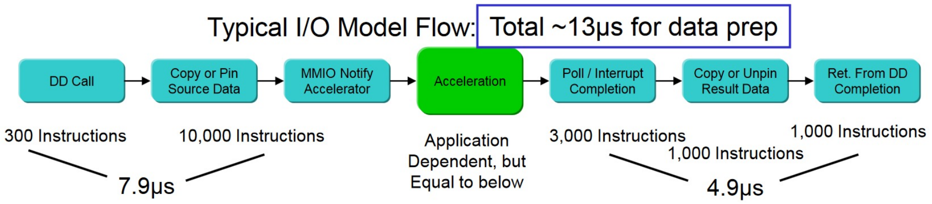
[3] https://github.com/open-power/amester

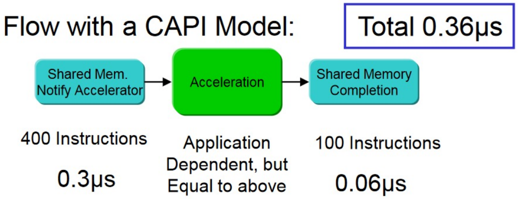
CAPI Technology Overview





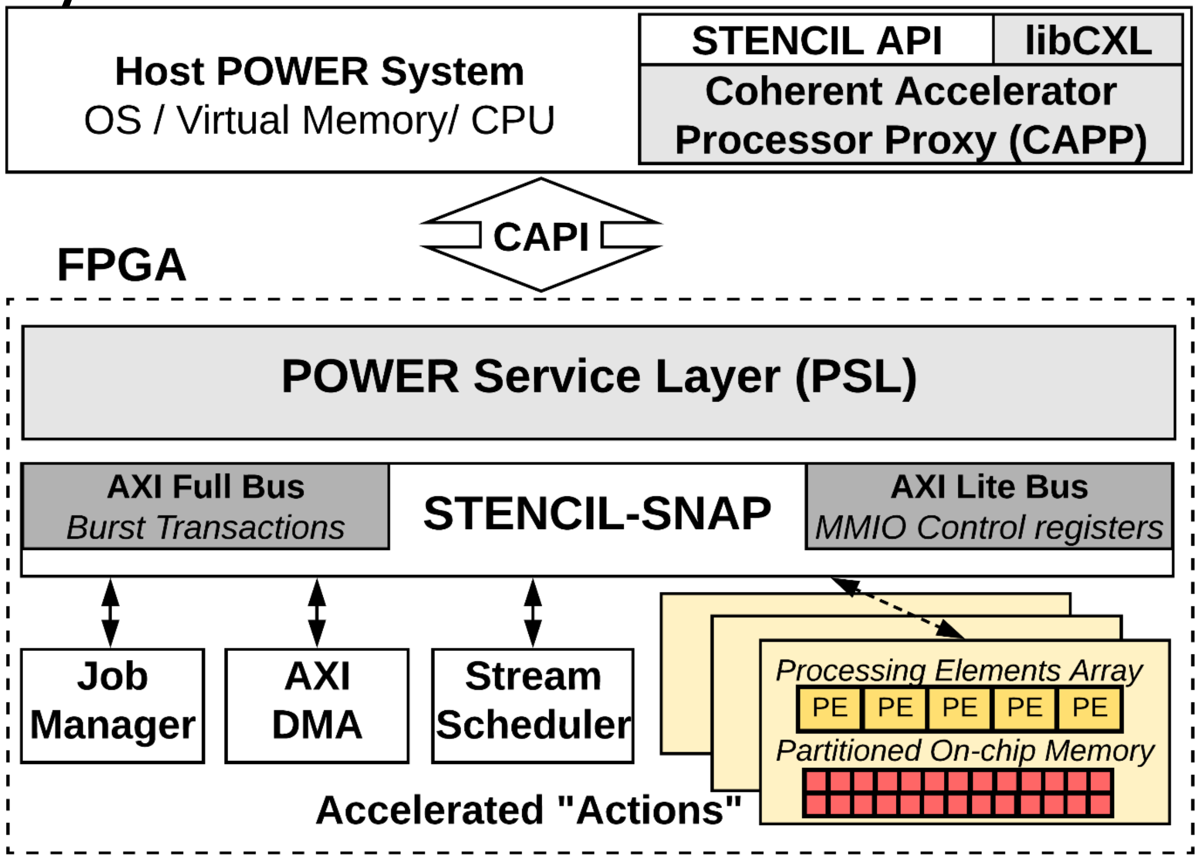






The Accelerator Architecture

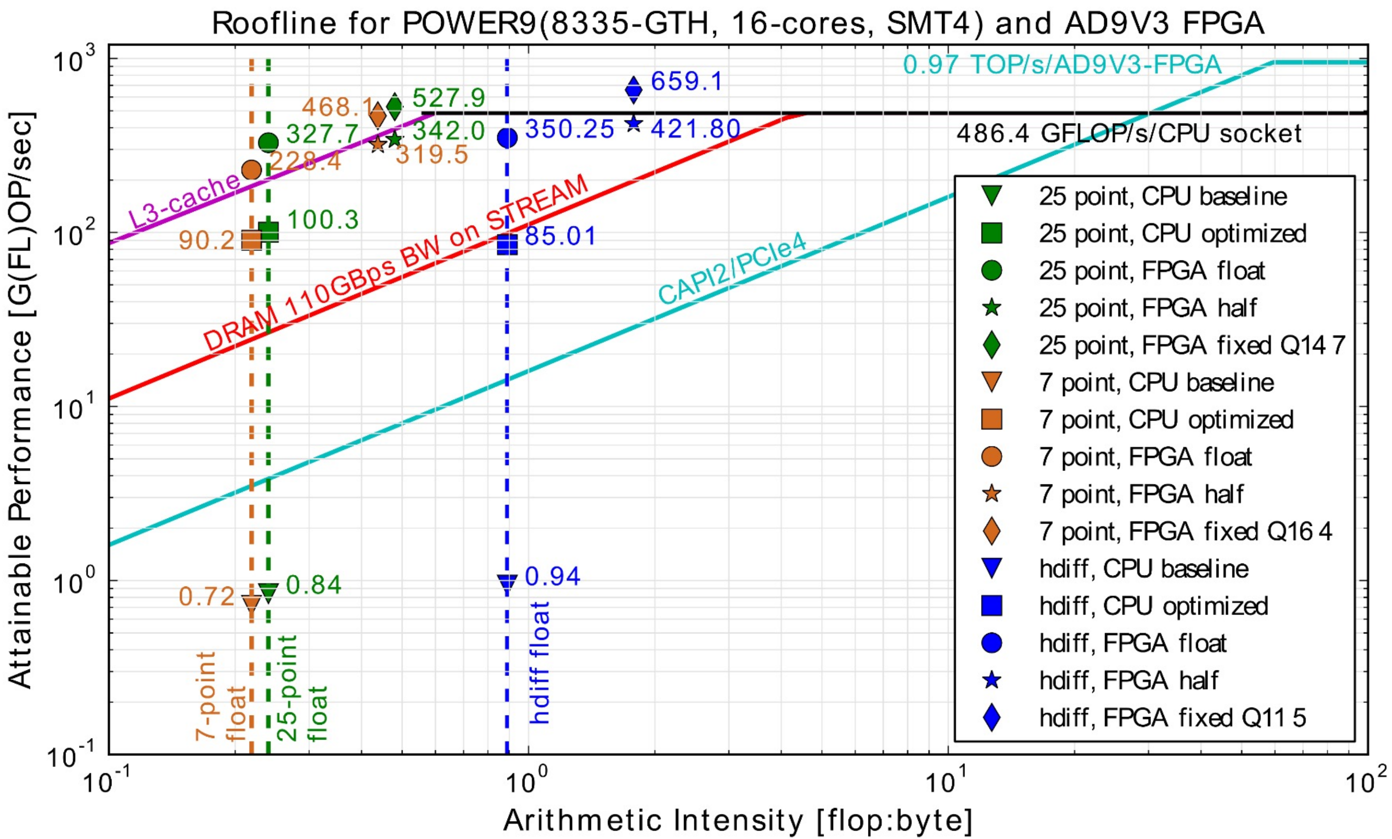
- Accelerators are acting as peers to CPU, by accessing the main memory through a high-performance cache-coherent link, enabled by PSL.



- Offloading jobs ("actions") to accelerators is handled by a software-defined API, with an interrupt-based queuing mechanism, allowing minimal CPU usage (thus power) during FPGA use.

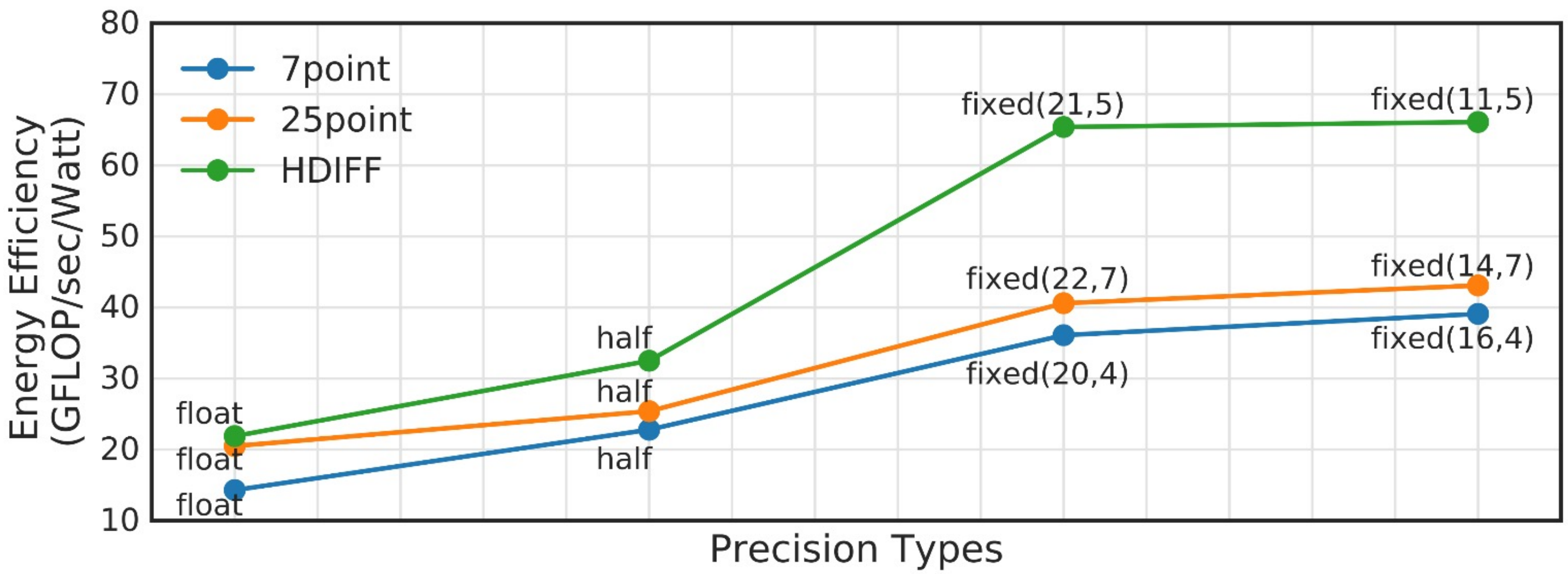
FPGA-aware Roofline

- Performance gap on multi-core bridged by exploiting data locality
- FPGA improves throughput by 2.5x – 4.1x compared to multi-core
- Using reduced-precision formats improves throughput by additional ~2x



FPGA energy-efficiency

- MPSoC to FPGA: 10x – 30x energy-efficiency
- Single-precision to half-precision float: reduced #DSPs and #BRAMs per FLOP
- Float to Fixed-point: significant reduction in #DSPs per FLOP
- Reducing bit-width further only reduces #BRAMs (#DSPs remain the same)



Conclusion and Summary

- **Motivation:** Low precision computing is a promising approach to solve data movement bottleneck for emerging big data workloads
- **Problem:** A key barrier to a widespread adoption of reduced-precision computing is the lack of an architecture exploiting arbitrary precision, supported by a software layer that controls the precision of computations.
- **Our contribution:**
  - Systematic precision exploration for various 3D stencils for a wide range of number systems-fixed, float, posit
  - Using state-of-the-art MPSoC with FPGA to show the capability of reduced precision
- **Evaluation**
  - **50% lower bits with only 1% loss of accuracy for all the number systems**
  - **Lower precision leads to ~FPGA peak performance of 468-659 GOP/s with 30-50x higher energy efficiency**

NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning

Executive Summary

- **Motivation:** A promising paradigm to alleviate data movement bottleneck is *near-memory computing (NMC)*, which consists of placing compute units close to the memory subsystem
- **Problem:** Simulation times are extremely slow, imposing long run-time especially in the early-stage design space exploration
- **Goal:** A quick high-level performance and energy estimation framework for NMC architectures
- **Our contribution: NAPEL**
  - Fast and accurate performance and energy prediction for previously-unseen applications using ensemble learning
  - Use intelligent statistical techniques and micro-architecture-independent application features to minimize experimental runs
- **Evaluation**
  - NAPEL is, on average, 220x faster than state-of-the-art NMC simulator
  - Error rates (average) of 8.5% and 11.5% for performance and energy estimation

We open source Ramulator-PIM: https://github.com/CMU-SAFARI/ramulator-pim/

SKA
300PB

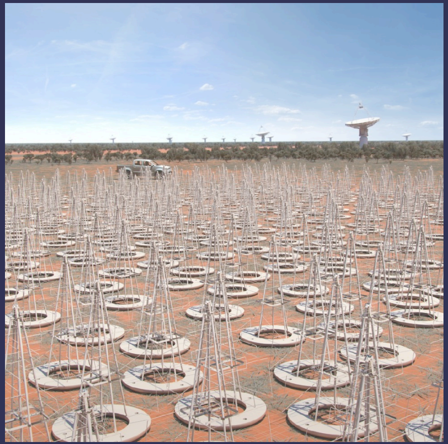
uploads on
facebook
180PB

searches on
Google
98PB

YouTube
15PB

CERN
15PB

NASDAQ
3PB



SKA
300PB

uploads on

Massive amounts of data

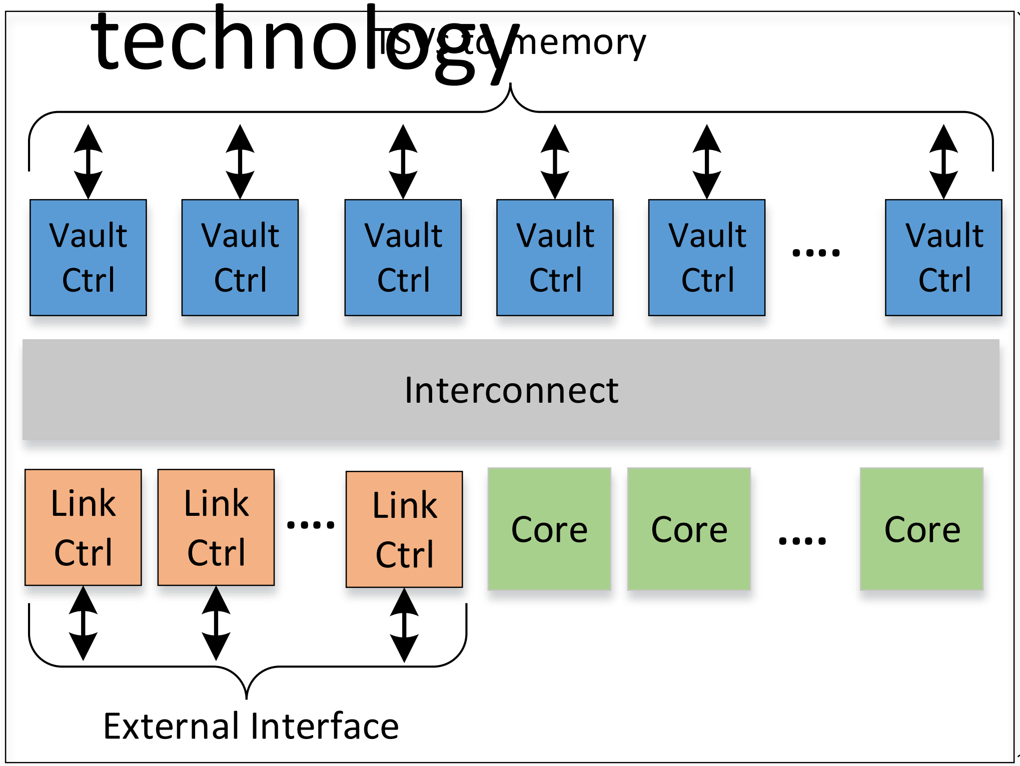
YouTube
15PB

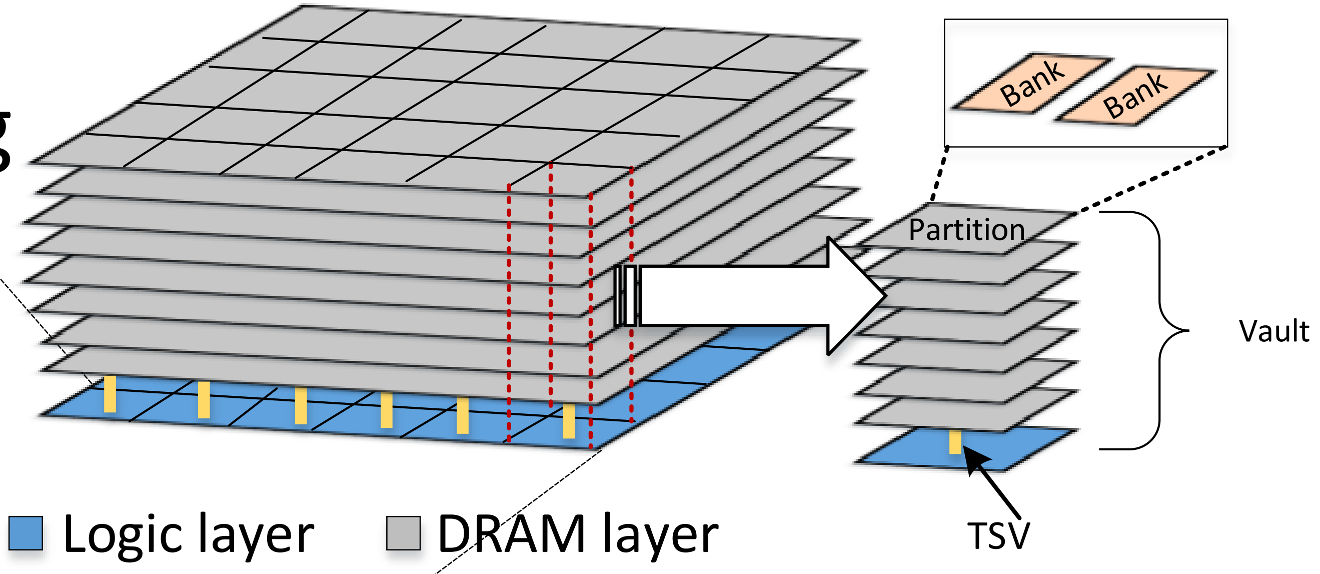
CERN
15PB

NASDAQ
3PB

Paradigm Shift - NMC

- Compute-centric to a data-centric approach
- Biggest enabler – stacking technology





NMC Simulators

- Simulation for:
  - Design space exploration (DSE)
  - Workload suitability analysis
- NMC Simulators:
  - Sinuca, 2015
  - HMC-SIM, 2016
  - CasHMC, 2016
  - Smart Memory Cube (SMC), 2016
  - CLAPPS, 2017
  - Gem5+HMC, 2017
  - Ramulator-PIM[1], 2019

[1]Ramulator-PIM: https://github.com/CMU-SAFARI/ramulator-pim/

NMC Simulators

- Simulation for:
  - Design space exploration (DSE)
  - Workload suitability analysis

Simulation of real workloads can be 10000x slower than native-execution!!!

  - Smart Memory Cube (SMC), 2016
  - CLAPPS, 2017
  - Gem5+HMC, 2017
  - Ramulator-PIM[1], 2019

[1]Ramulator-PIM: https://github.com/CMU-SAFARI/ramulator-pim/

NMC Simulators

- Simulation for:
  - Design space exploration (DSE)
  - Workload suitability analysis

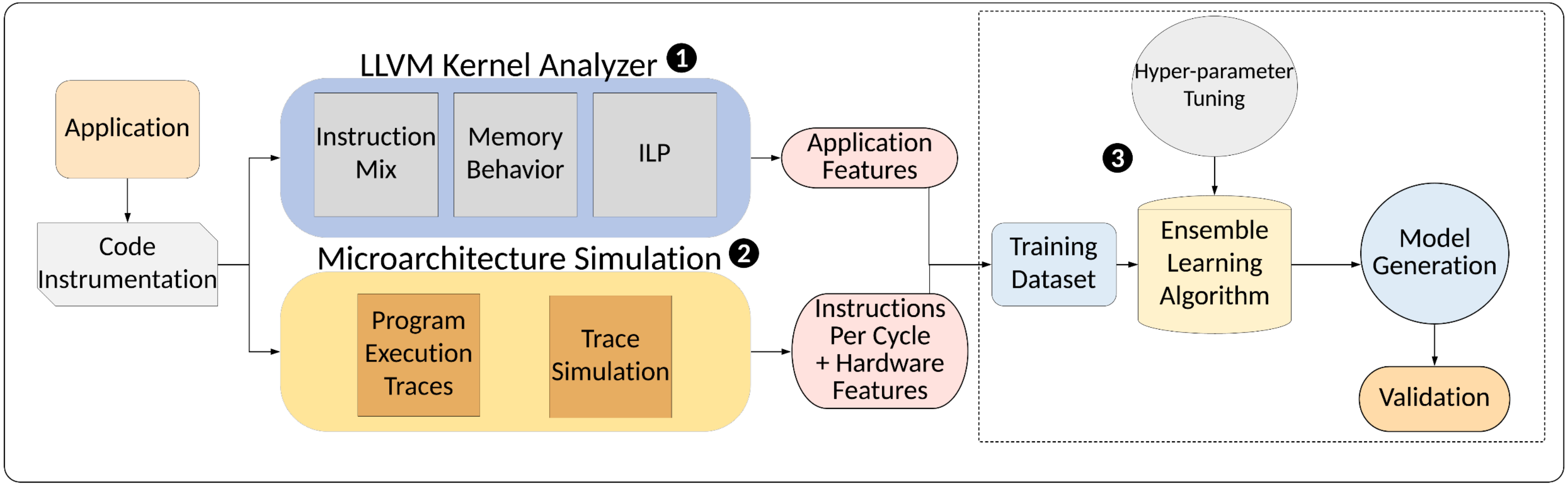
Idea: Leverage ML with statistical techniques for quick NMC performance/energy prediction

  - Smart Memory Cube (SMC), 2016
  - CLAPPS, 2017
  - Gem5+HMC, 2017
  - Ramulator-PIM[1], 2019

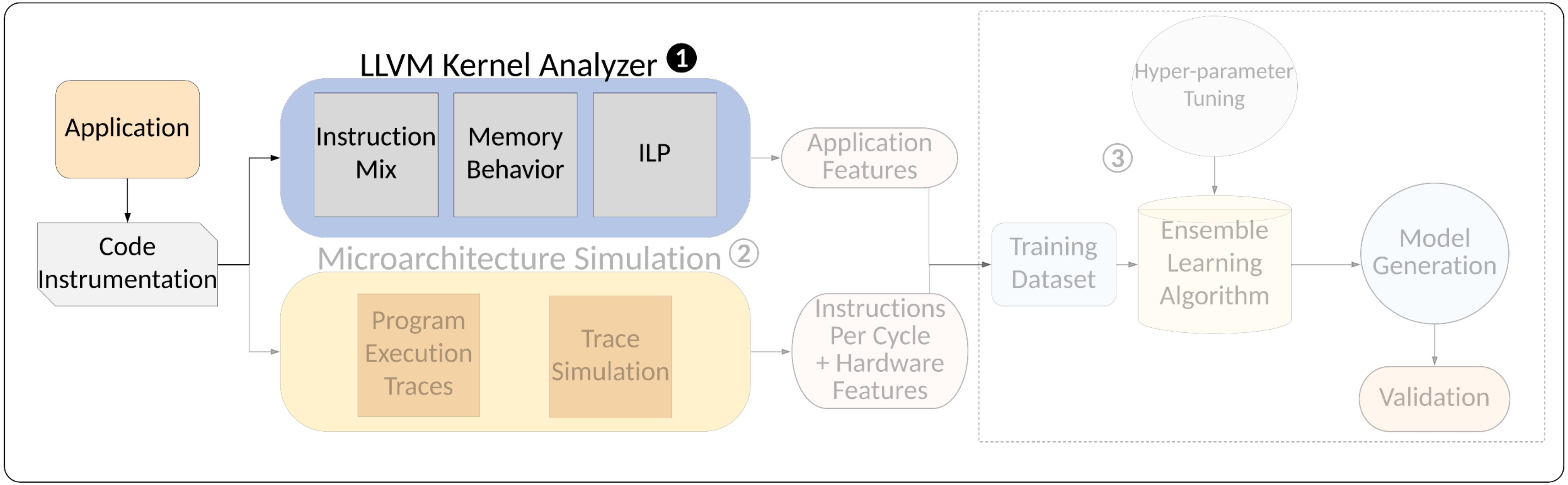
[1]Ramulator-PIM: https://github.com/CMU-SAFARI/ramulator-pim/

NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning

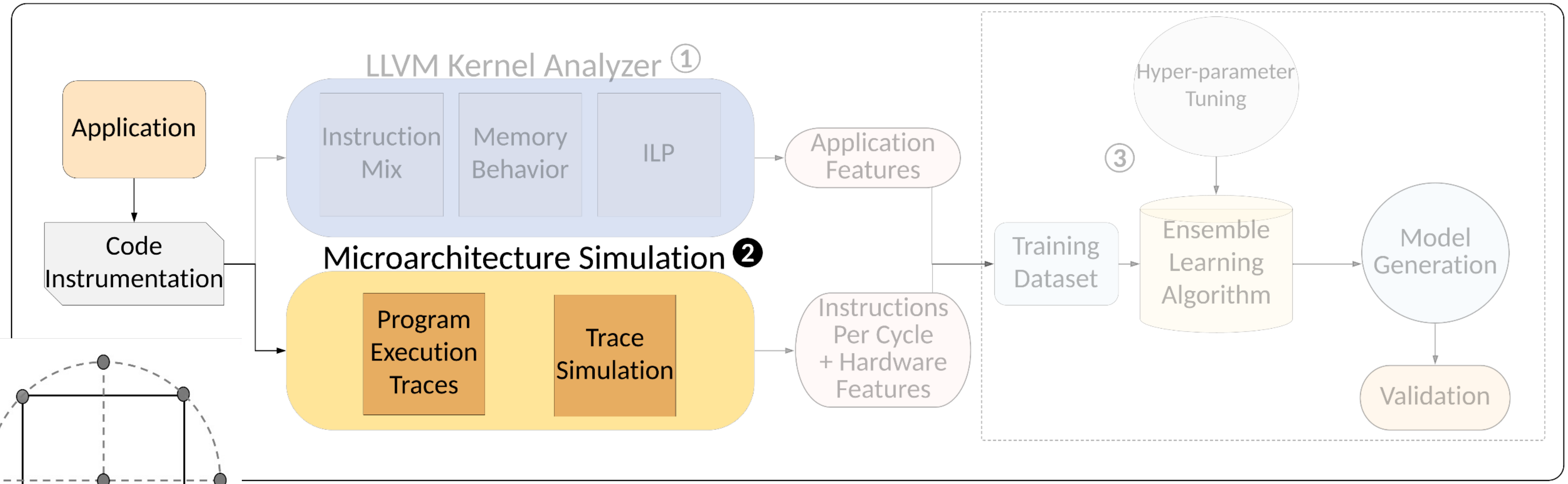
NAPEL Model Training

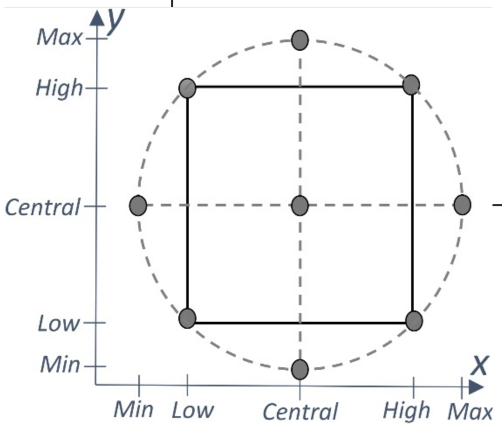


Phase 1: LLVM Analyzer



Phase 2: Microarchitecture Simulation

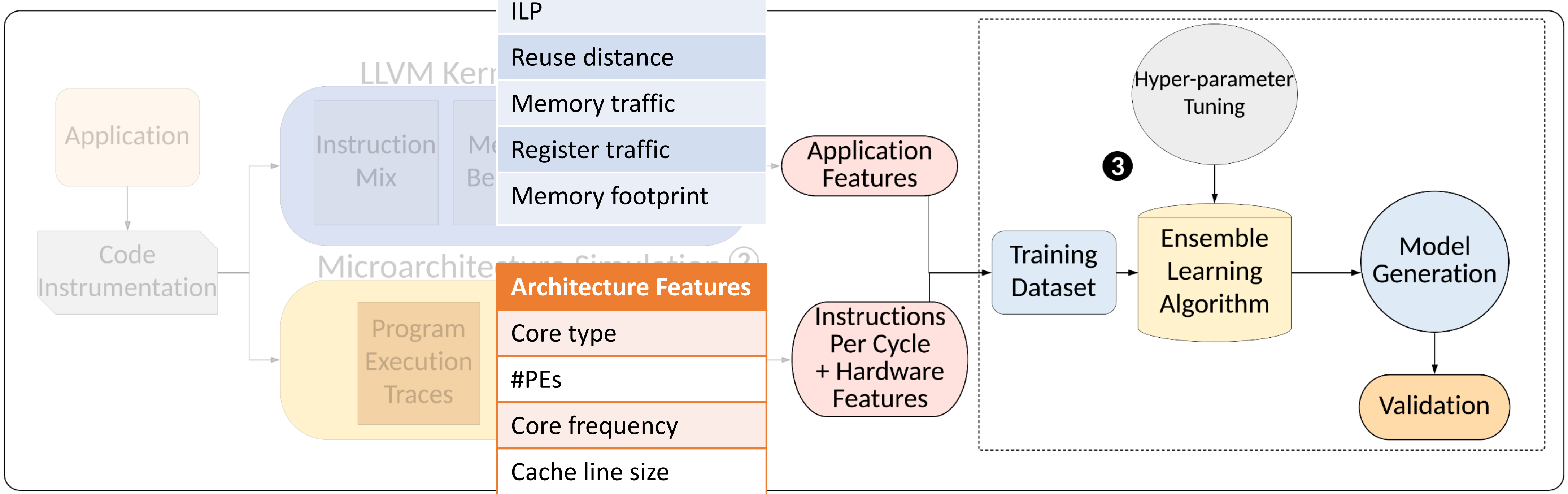




Central composite design of experiments technique to minimize the number of experiments while data collection

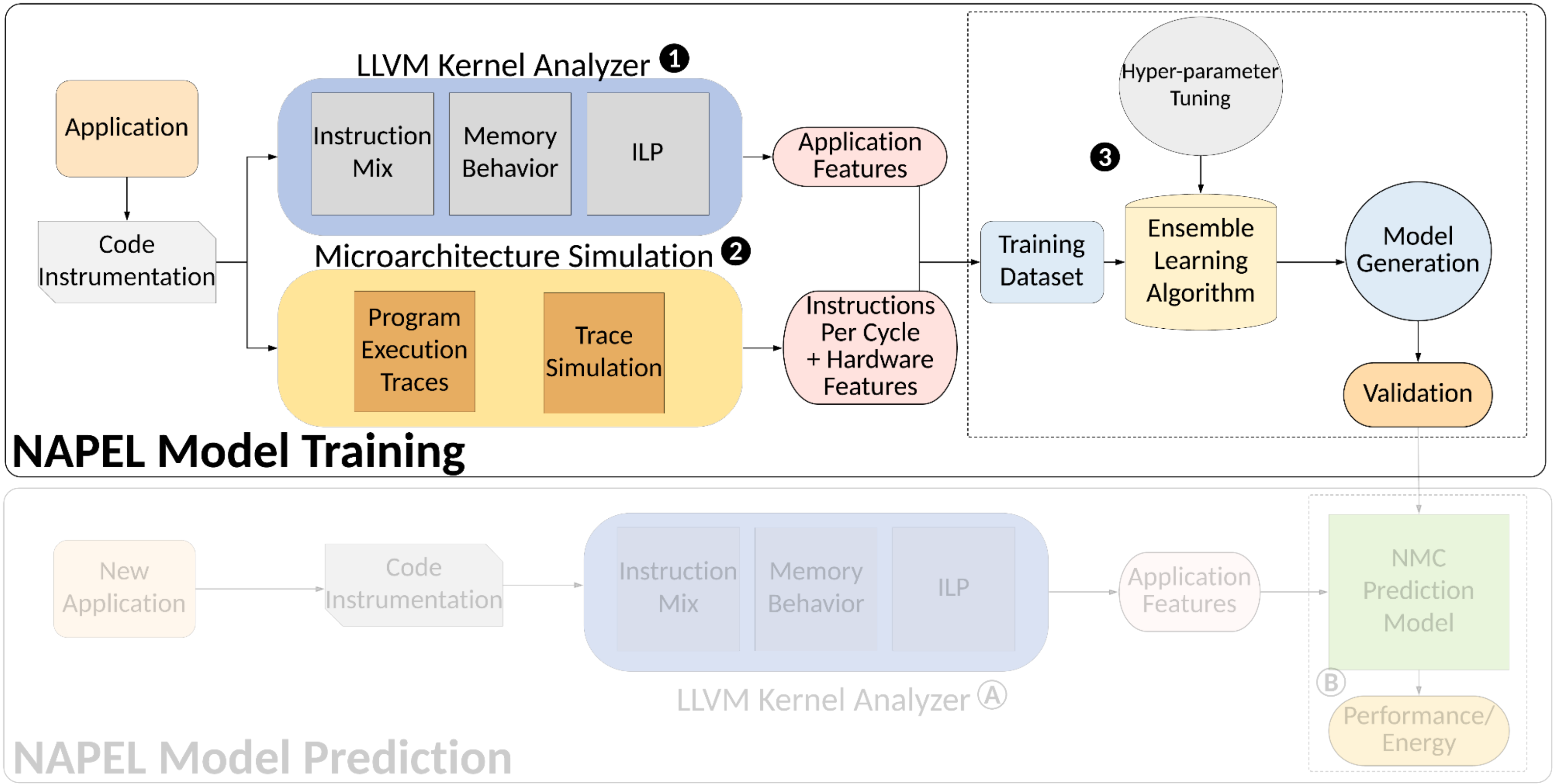
Phase 3: Ensemble ML Training

Application Features
Instruction Mix
ILP
Reuse distance
Memory traffic
Register traffic
Memory footprint

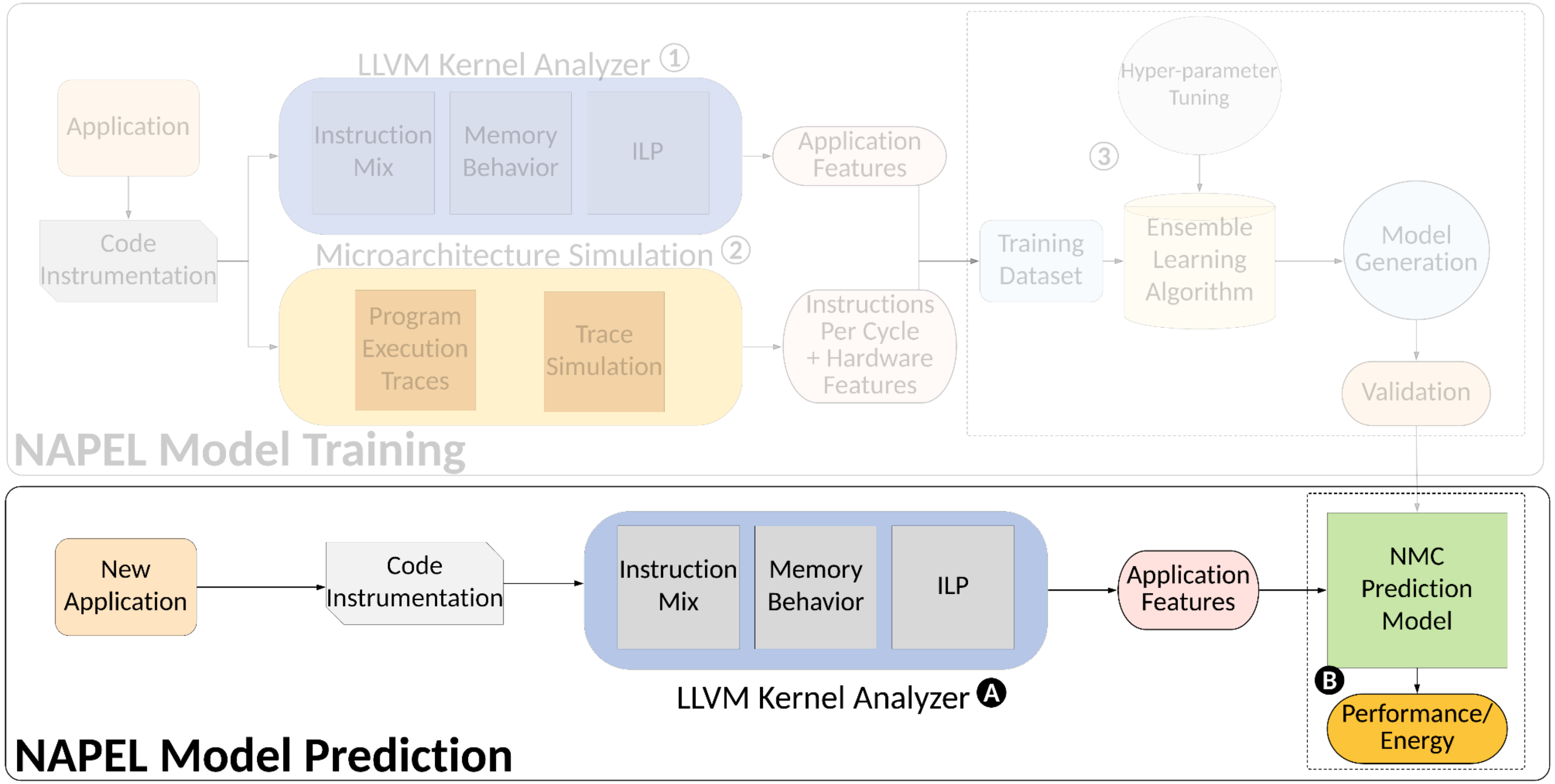


Architecture Features
Core type
#PEs
Core frequency
Cache line size
DRAM layers
Cache access fraction
DRAM access fraction

NAPEL Framework

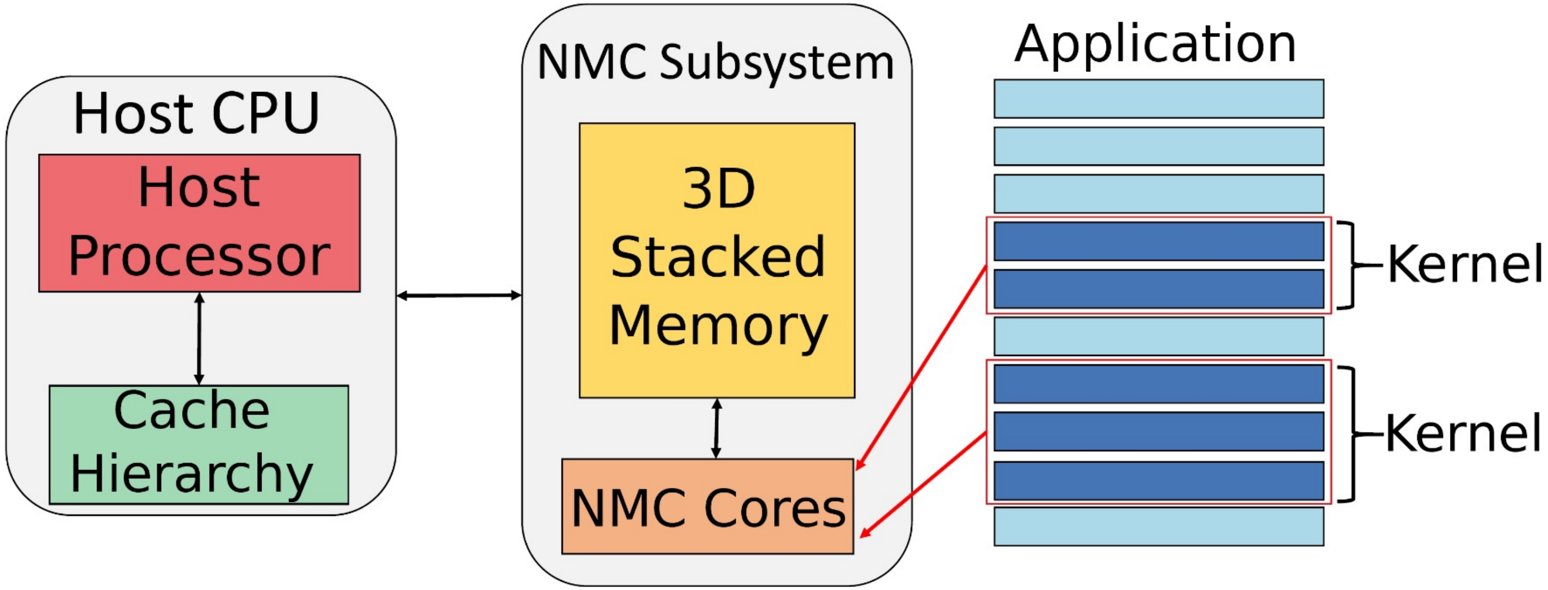


NAPEL Prediction



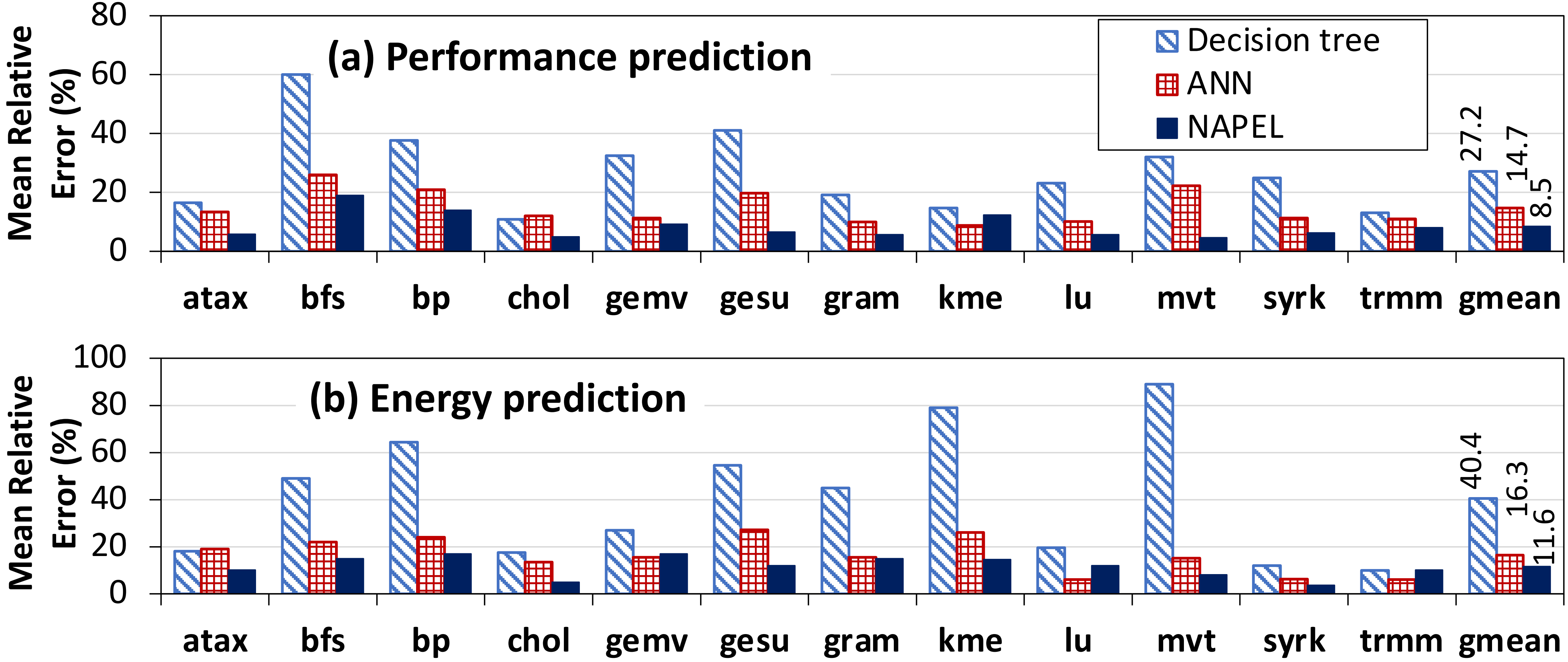
Experimental Setup

- Host System
  - **IBM POWER9**
  - Power: AMESTER
- NMC Subsystem
  - **Ramulator-PIM**[1]
- Workloads
  - **PolyBench** and **Rodinia**
  - Heterogeneous workloads such as image processing, machine learning, graph processing etc.
- Accuracy in terms of mean relative error (MRE)

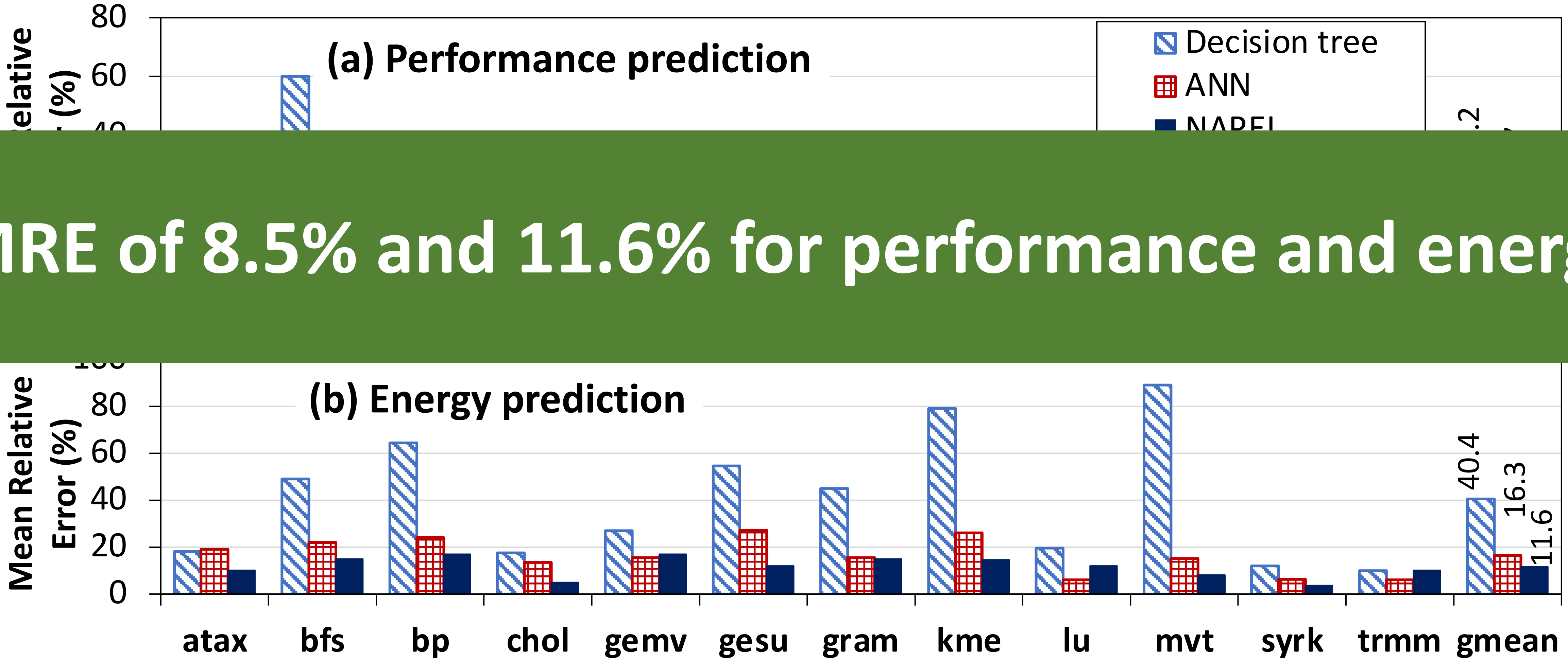


[1]https://github.com/CMU-SAFARI/ramulator-pim/

NAPEL Accuracy: Performance and Energy Estimates



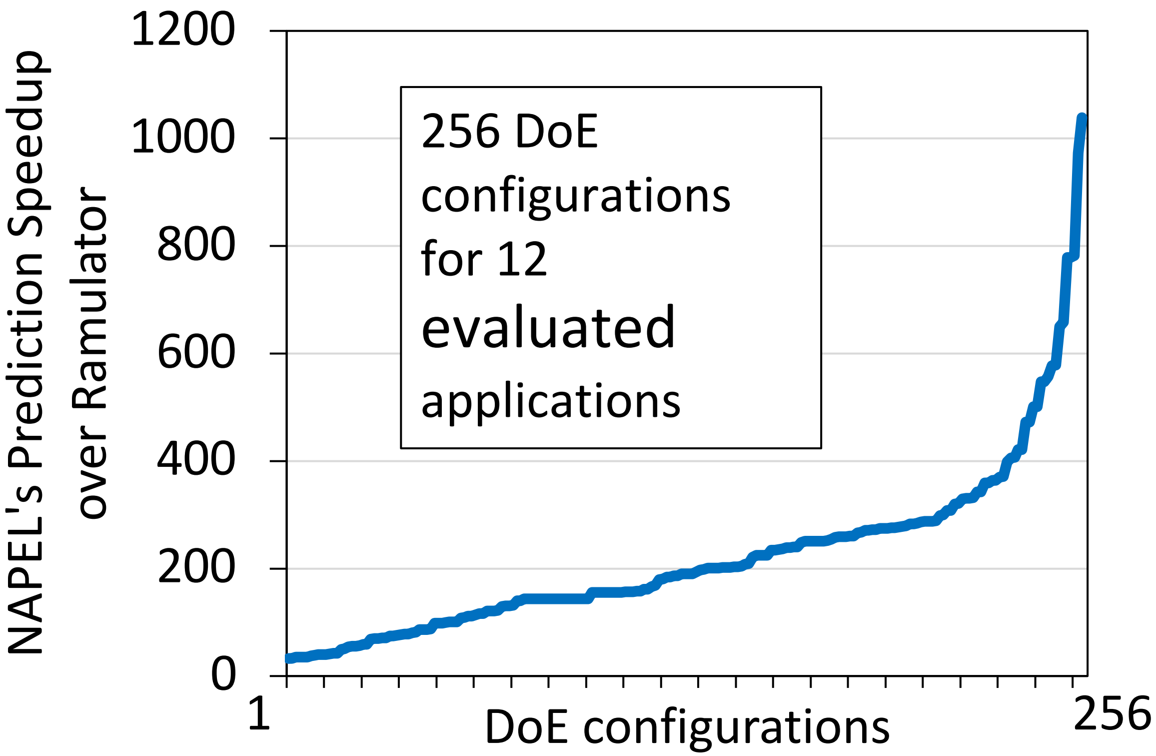
NAPEL Accuracy: Performance and Energy Estimates



MRE of 8.5% and 11.6% for performance and energy

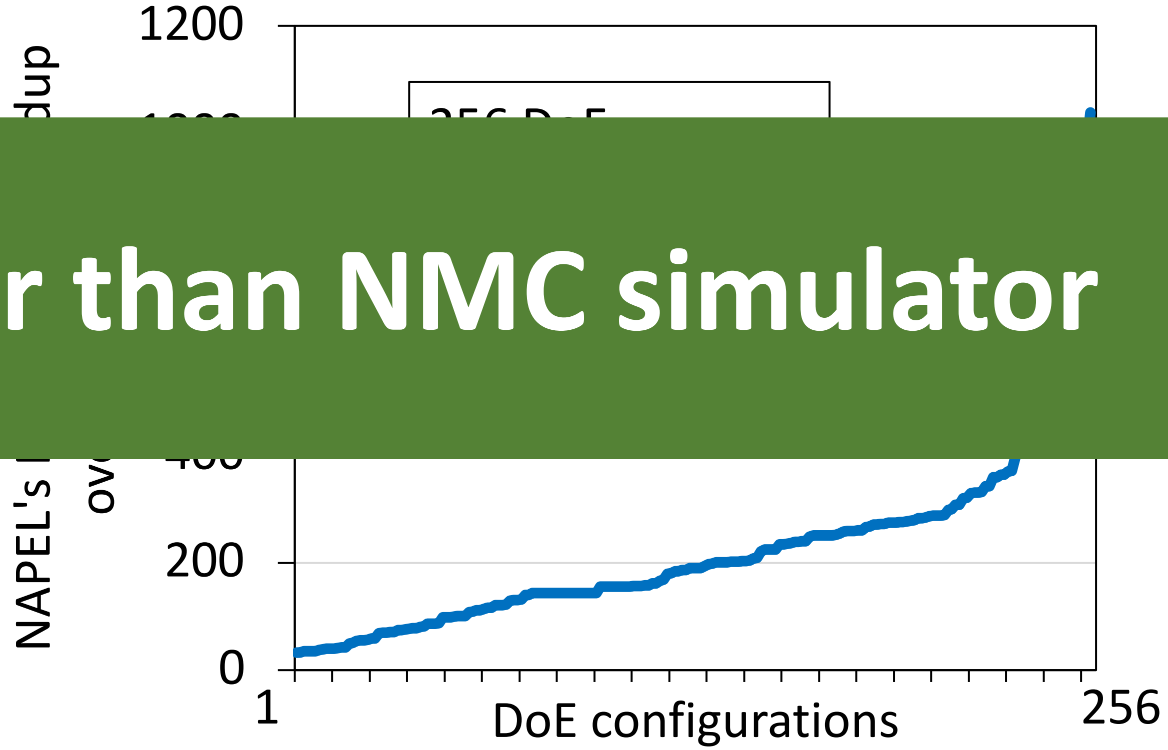
Speed of Evaluation

Application Name	Training/Prediction Time #DoE conf.	DoE run (mins)	Train+Tune (mins)	Pred. (mins)
atax	11	522	34.9	0.49
bfs	31	1084	34.2	0.48
bp	31	1073	43.8	0.47
chol	19	741	34.9	0.49
gemv	19	741	24.4	0.51
gesu	19	731	36.1	0.51
gram	19	773	36.5	0.52
kme	31	742	36.9	0.55
lu	19	633	37.9	0.51
mvt	19	955	38.0	0.54
syrk	19	928	35.7	0.51
trmm	19	898	37.6	0.48



Speed of Evaluation

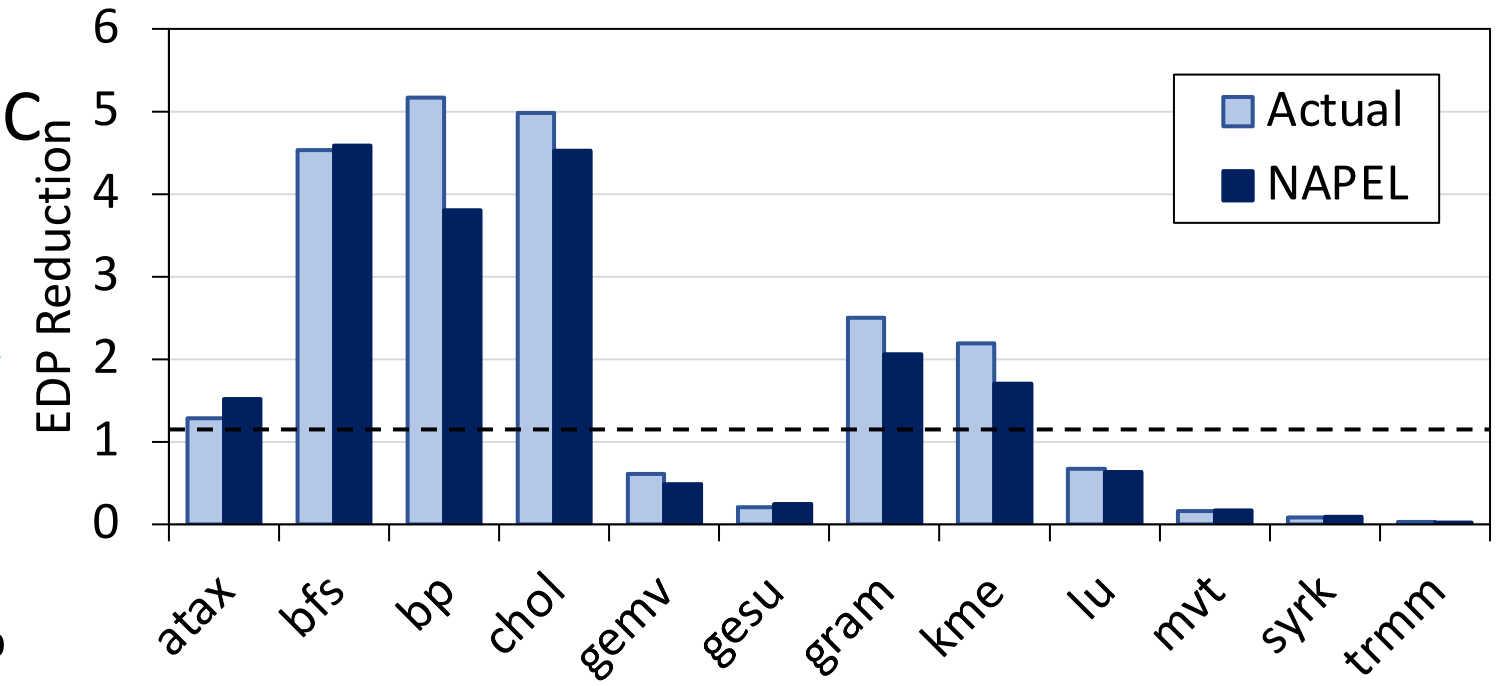
Application	Training/Prediction Time			
kme	31	742	36.9	0.55
lu	19	633	37.9	0.51
mvt	19	955	38.0	0.54
syrk	19	928	35.7	0.51
trmm	19	898	37.6	0.48



220x (up to 1039x) faster than NMC simulator

Use Case: NMC Suitability Analysis

- Assess the potential of offloading a workload to NMC
- NAPEL provides accurate prediction of NMC suitability
- MRE between 1.3% to 26.3% (average 14.1%)



Conclusion and Summary

- **Motivation:** A promising paradigm to alleviate data movement bottleneck is *near-memory computing (NMC)*, which consists of placing compute units close to the memory subsystem
- **Problem:** Simulation times are extremely slow, imposing long run-time especially in the early-stage design space exploration
- **Goal:** A quick high-level performance and energy estimation framework for NMC architectures
- **Our contribution: NAPEL**
  - Fast and accurate performance and energy prediction for previously-unseen applications using ensemble learning
  - Use intelligent statistical techniques and micro-architecture-independent application features to minimize experimental runs
- **Evaluation**
  - NAPEL is, on average, 220x faster than state-of-the-art NMC simulator
  - Error rates (average) of 8.5% and 11.5% for performance and energy estimation

We open source Ramulator-PIM: https://github.com/CMU-SAFARI/ramulator-pim/

LEAPER:
Modeling Cloud FPGA-based systems via transfer learning

Executive Summary

Motivation: Machine-learning-based models have gained traction to overcome the slow downstream implementation process of FPGAs.

Problem: (1) A model trained for a specific environment cannot predict for a new, unknown environment (2) Training requires large amounts of data, which is cost-inefficient because of the time-consuming FPGA design cycle.

Goal: Leverage and transfer our ML-based performance models trained on a low-end local system to a new, unknown, high-end FPGA-based system, thereby avoiding the aforementioned two main limitations of traditional ML-based approaches.

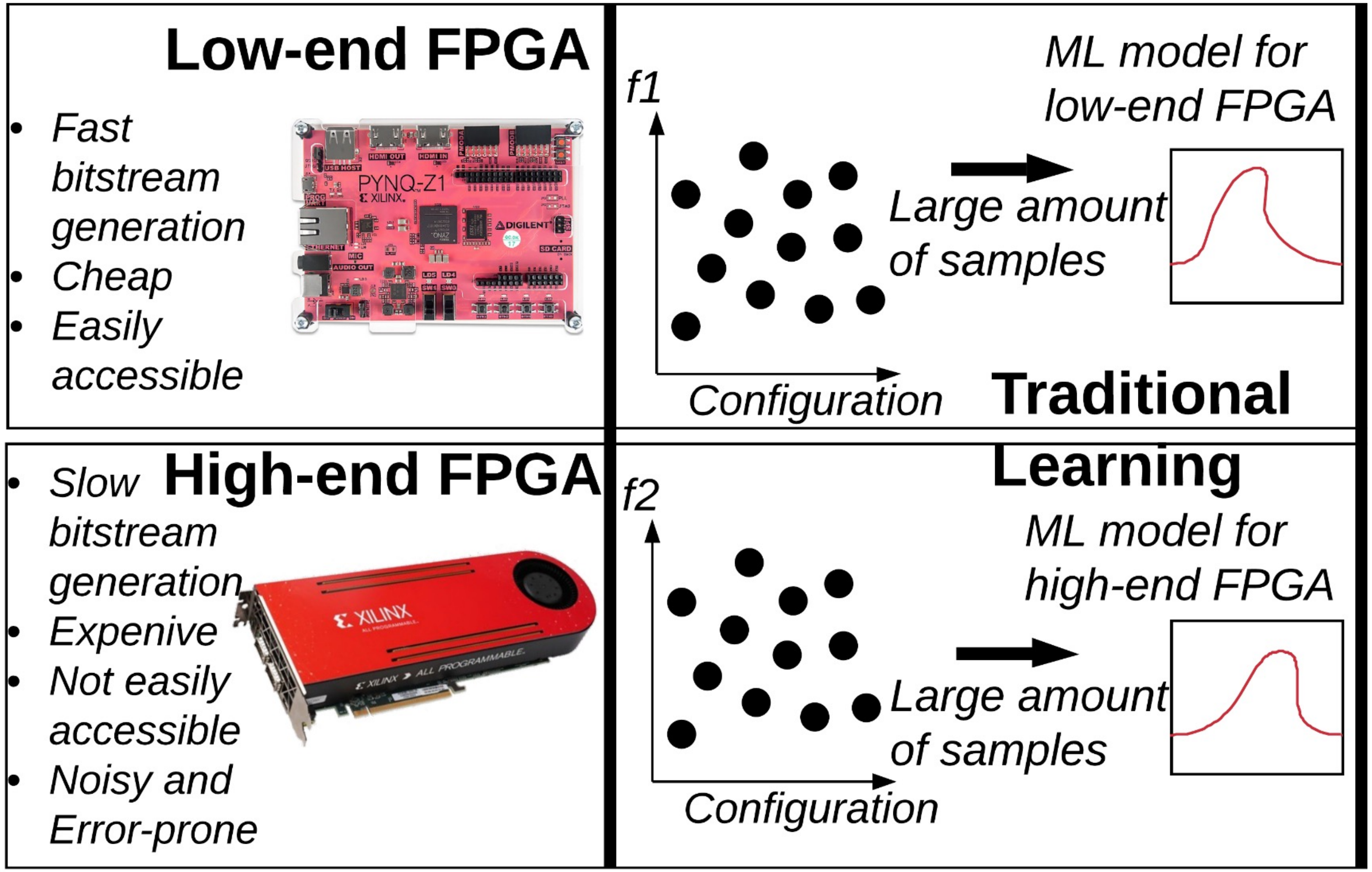
Our contribution:

- First **transfer learning-based** approach for FPGA-based systems that allows us to leverage a model trained on a **low-end edge** FPGA and adapt it to **high-end** FPGA-based systems via **few-shot learning.**

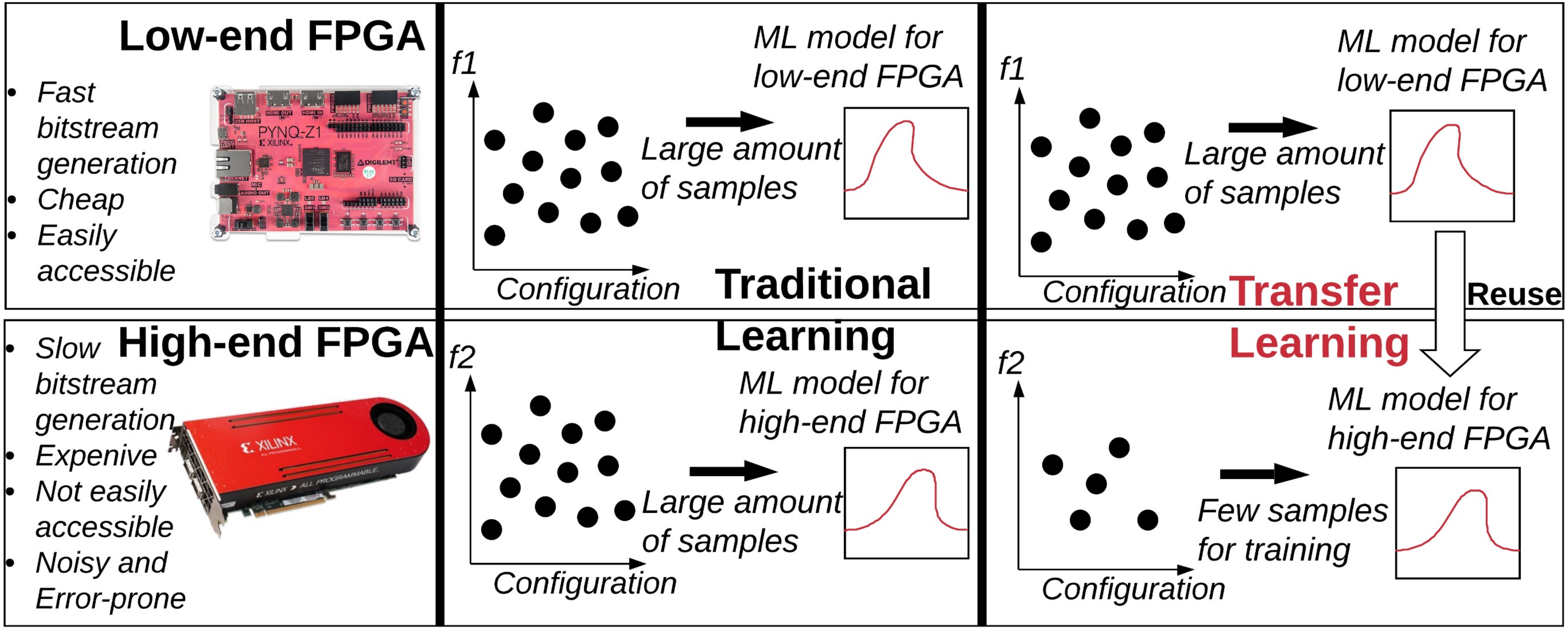
Evaluation

- Demonstrate our approach across five state-of-the-art, high-end FPGA-based platforms with three different interconnect technologies on six real-world applications.
- Transferred models from a low-end edge board to high-end FPGA-based systems achieve high accuracy of 80-90% for resource prediction.

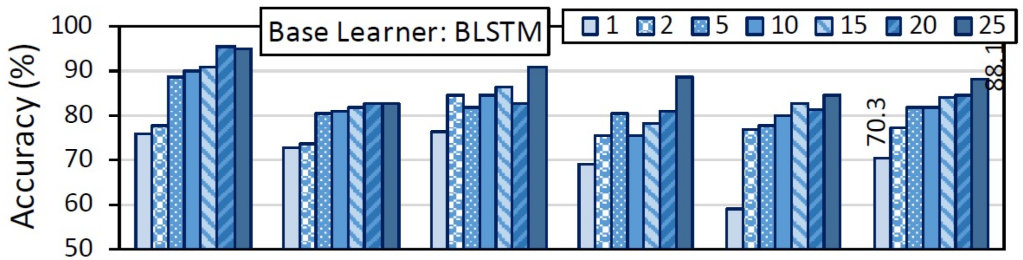
Traditional Approach

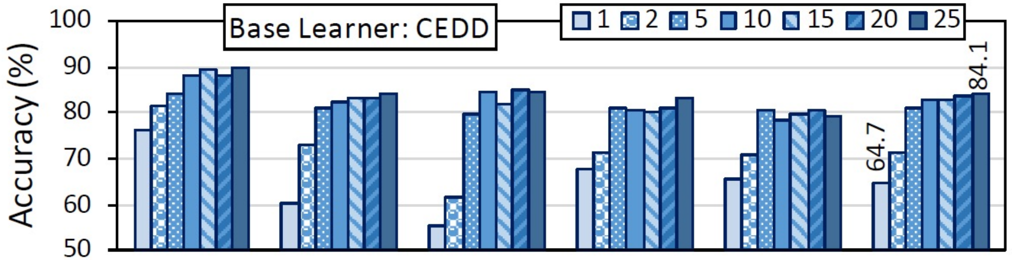


Our Approach

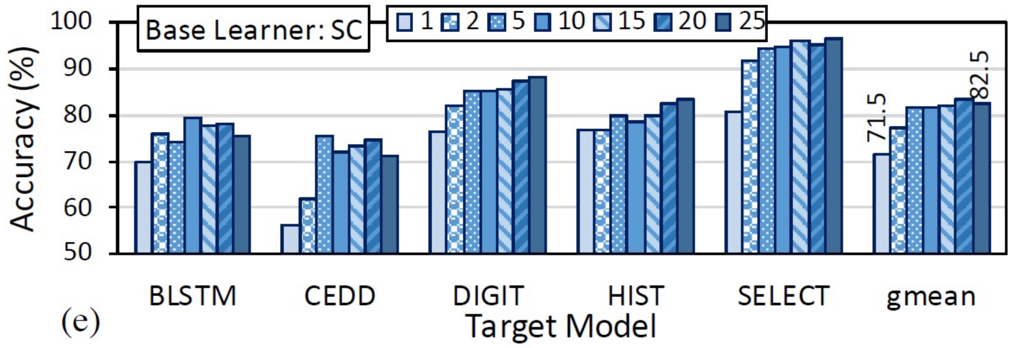


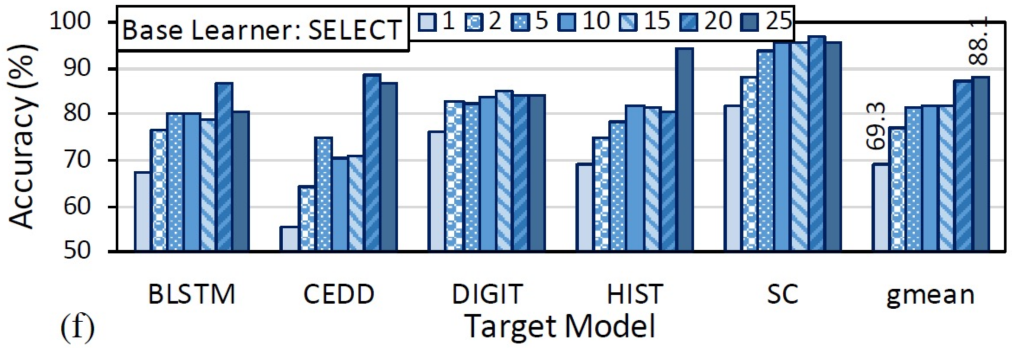
Results: Resource Model Transfer





Transferred models achieve high accuracy of 80-90% for resource prediction





Complete List of Publications

1. **Gagandeep Singh,** Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gomez-Luna, Henk Corporaal, and Onur Mutlu, "FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications", IEEE Micro 2021

2. **Gagandeep Singh**, Dionysios Diamantopoulos, Juan Gomez-Luna, Sander Stuijk, Onur Mutlu and Henk Corporaal, "Modeling FPGA-Based Heterogeneous Computing via Few-Shot Learning", FPGA 2021

3. **Gagandeep Singh**, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gomez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal, "NERO: A Near-High Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling", FPL 2020

4. **Gagandeep Singh**, Juan Gómez-Luna, Giovanni Mariani, Geraldo F. Oliveira, Stefano Corda, Sander Stuijk, Onur Mutlu, and Henk Corporaal, "NAPEL: Near-memory computing application performance prediction via ensemble learning." DAC 2019

5. **Gagandeep Singh**, Dionysios Diamantopoulos, Christoph Hagleitner, Sander Stuijk, and Henk Corporaal, "NARMADA: Near-memory horizontal diffusion accelerator for scalable stencil computations." FPL 2019

6. **Gagandeep Singh**, Dionysios Diamantopoulos, Sander Stuijk, Christoph Hagleitner, and Henk Corporaal, "Low precision processing for high order stencil computations." LNCS 2019

7. **Gagandeep Singh,** Lorenzo Chelini, Stefano Corda, Ahsan Javed Awan, Sander Stuijk, Roel Jordans, Henk Corporaal, and Albert-Jan Boonstra, "Near-memory computing: Past, present, and future." MICPRO 2019

8. **Gagandeep Singh**, Lorenzo Chelini, Stefano Corda, Ahsan Javed Awan, Sander Stuijk, Roel Jordans, Henk Corporaal, and Albert-Jan Boonstra, "A Review of Near Memory Computing Architectures Opportunities and Challenges." DSD 2019

9. Dionysios Diamantopoulos, Burkhard Ringlein, Mitra Purandare, **Gagandeep Singh,** and Christoph Hagleitner, "Agile Autotuning of a Transprecision Tensor Accelerator Overlay", FPL 2020

10. Kanishkan Vadivel, Lorenzo Chelini, Ali Bana Gozar, **Gagandeep Singh**, Stefano Corda, Roel Jordans and Henk Corporaal, "TDO-CIM: Transparent Detection and Offloading for Computation In-memory", DATE 2020

11. Corda, Stefano, **Gagandeep Singh**, Ahsan Javed Awan, Roel Jordans, and Henk Corporaal, "Memory and parallelism analysis using a platform-independent approach." SCOPES 2019

12. Corda, Stefano, **Gagandeep Singh**, Ahsan Javed Awan, Roel Jordans, and Henk Corporaal, "Platform independent software analysis for near memory computing." DSD 2019.

13. Jan van Lunteren, Ronald Luijten, Dionysios Diamantopoulos, Florian Auernhammer, Christoph Hagleitner, Lorenzo Chelini, Stefano Corda, **Gagandeep Singh,** "Coherently Attached Programmable Near-Memory Acceleration Platform and its application to Stencil Processing.", DATE 2019

Patent:

- Ronald Luijten, **Gagandeep Singh,** Joost VandeVondele, "CGRA accelerator for weather/climate dynamics simulation" P201909001US01