



ModSim in the AI Era: Quantitative Tools of Codesign

Adolfy Hoisie, Chair, Computing for National Security Department

with Lingda Li, Tom Flynn, Ray Ren, many others

Multicore World 2024

February 2024, Christchurch, New Zealand



Outline

- Challenges of Performance Modeling
- AI-based Modeling and Simulation
- Dynamic Modeling
- AI-based ModSim of Complex Scientific workflows
- Summary and Conclusions

Large-scale Experimental Facilities

Relativistic Heavy Ion Collider (**RHIC**): Supports more than 1000 scientists worldwide

RHIC



National Synchrotron Light Source II (**NSLS-II**): Newest and brightest synchrotron in the world; supports a multitude of scientific research in academia, industry, and national security

NSLS-II



Center for Functional Nanomaterials (**CFN**): Combines theory and experiment to probe materials

CFN



Accelerator Test Facility (**ATF**)

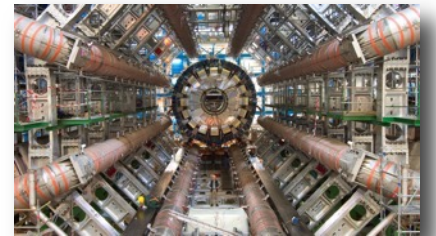
Large Hadron Collider (**LHC**) ATLAS: Largest Tier-1 center outside of CERN

Atmospheric Radiation Measurement (**ARM**) program: Partner in multi-site facility, operating its external data center

Belle II: Tier 0 computing for neutrino experiment

Quantum chromodynamics (**QCD**) computing facilities for Brookhaven Lab, RIKEN, and U.S. QCD communities

ATLAS



QCD



Brookhaven Lab Data by the Numbers

One of the top-10 scientific archives in the world*

- ~230 PB of data archived (exabytes by 2026)
- 26 million files injected (30 PB)
- 21 million files restored (34 PB)

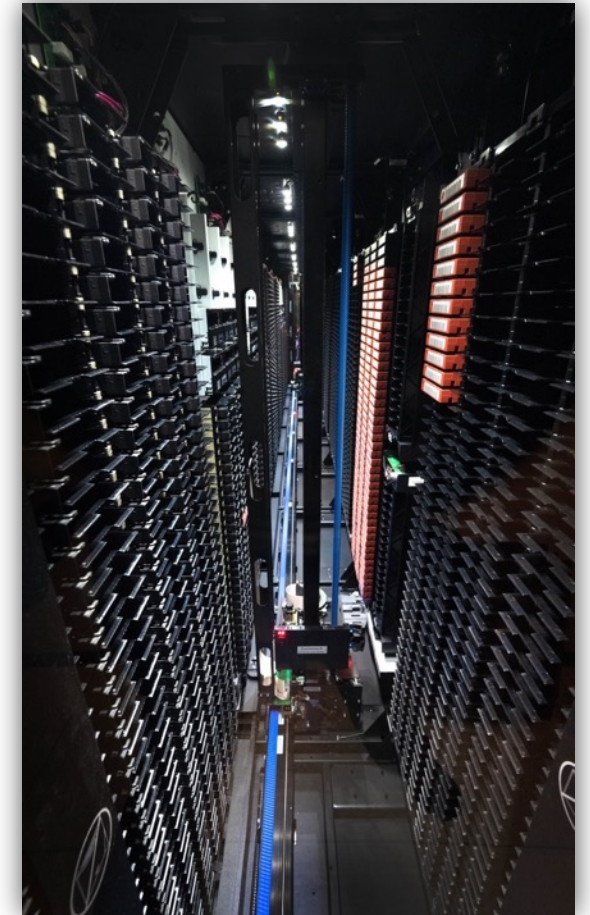
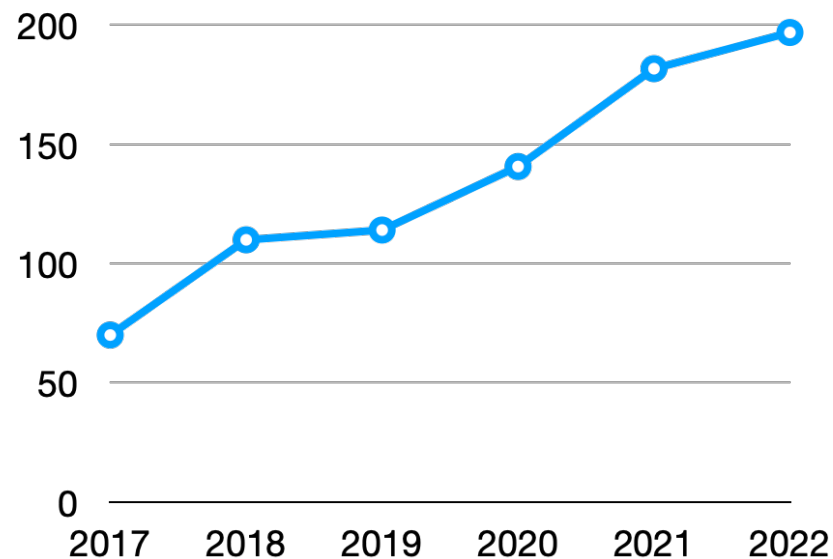
1.4 EB of data analyzed

200 PB of data transferred

- Data import: 90 PB
- Data export: 110 PB

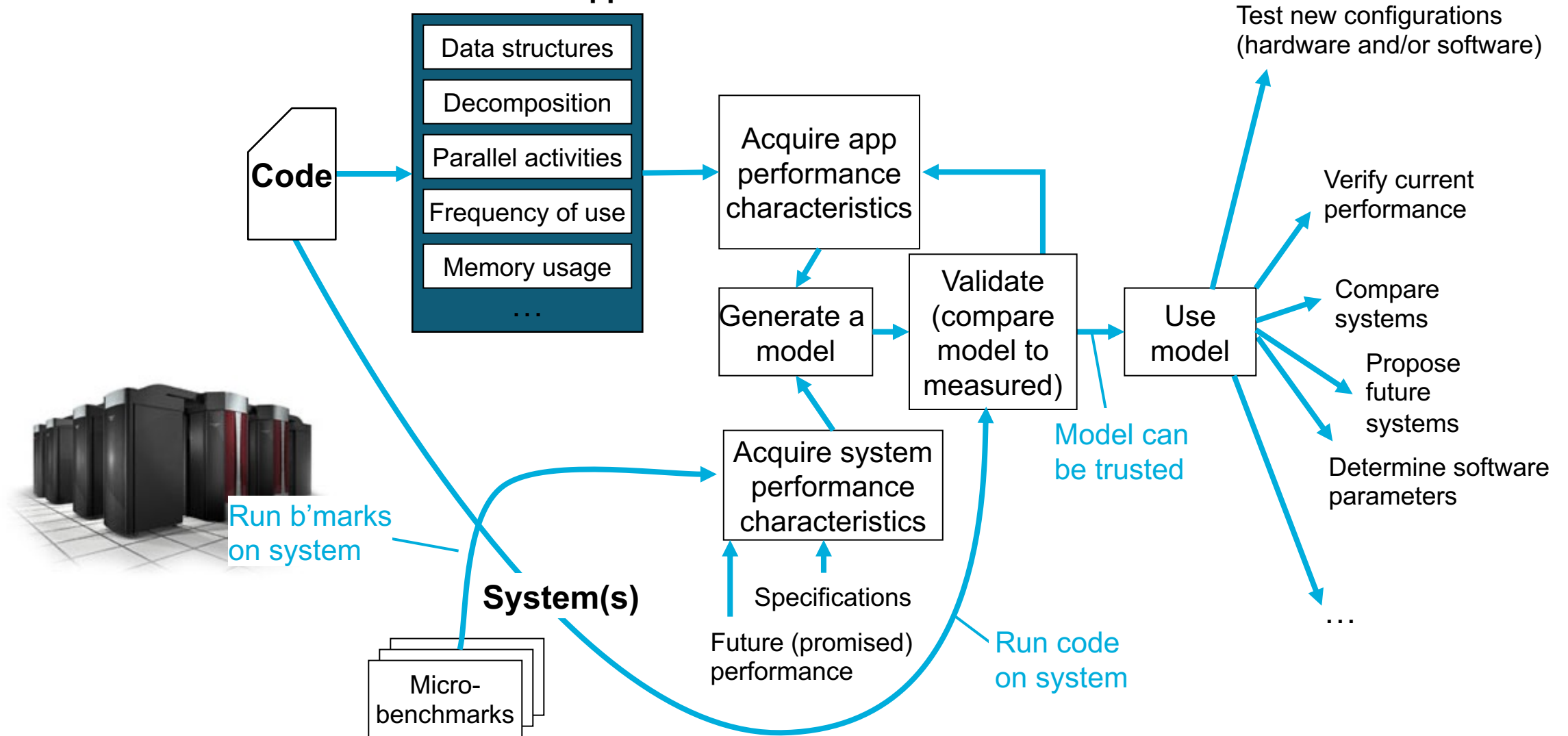
1,900 active accounts

2022 Statistics
BNL WAN Traffic/year [PB]



Performance Modeling Process Flow

Identification of application characteristics



Performance Prediction Methods: Speed versus Accuracy

Smart Modeling and Simulation for HPC (SMaSH) is an intricate challenge because of the complexity of the design space.








Methodologies exist that lack either practicality or accuracy.

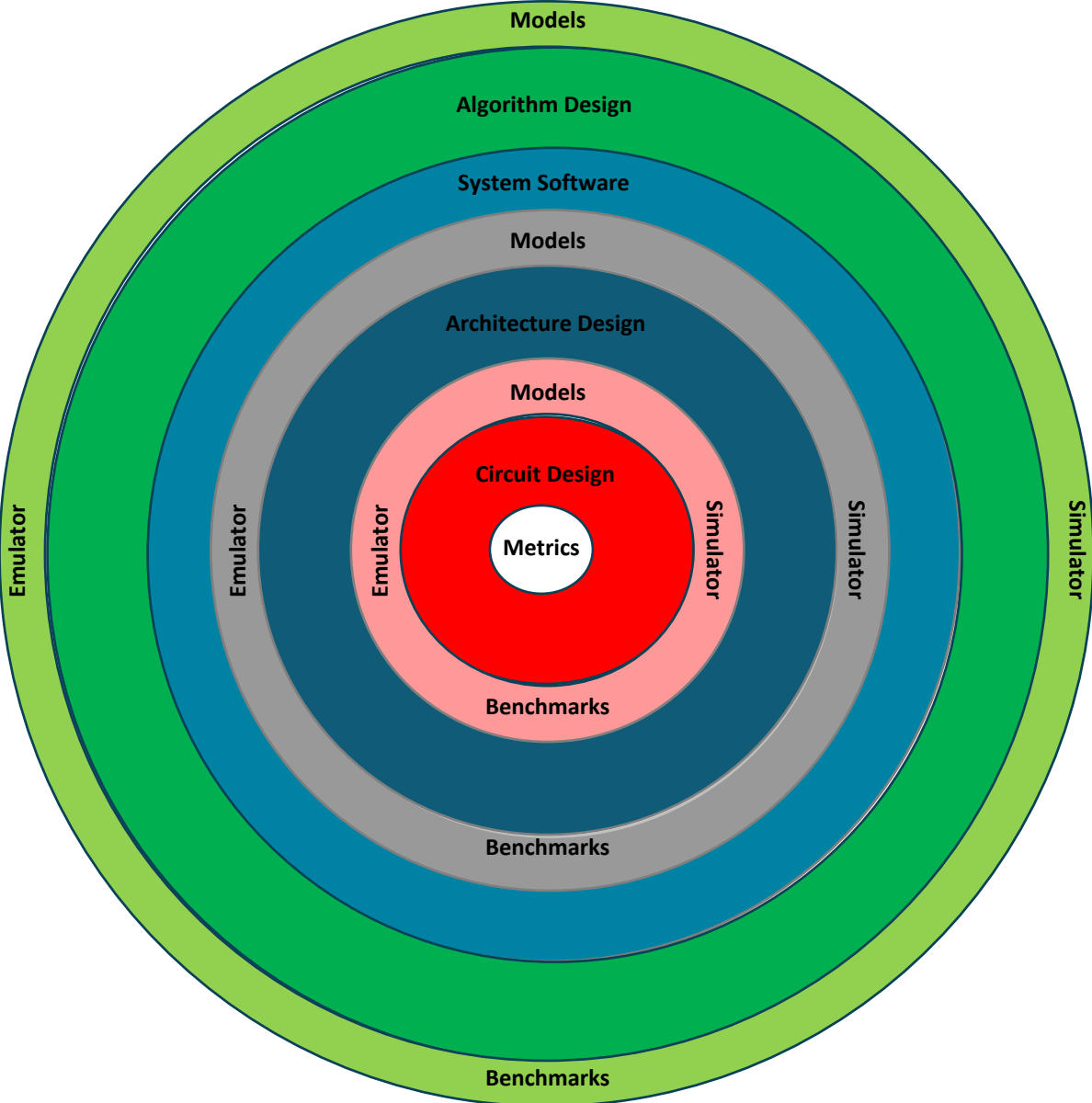
	Speed	Accuracy	Flexibility
Analytical Methodologies	Fast	Low	Low
Emulation	Fast	High (?)	Very low
Discrete Event Simulation	Slow	High	High
Machine Learning (ML)-based Simulation	Medium; aiming high	High	Medium; aiming high

Discrete event (DE) simulation is slow:

- For example, gem5 simulates a modern microprocessor at several hundreds of KIPS.
- Not practical for realistic architectures and workloads.

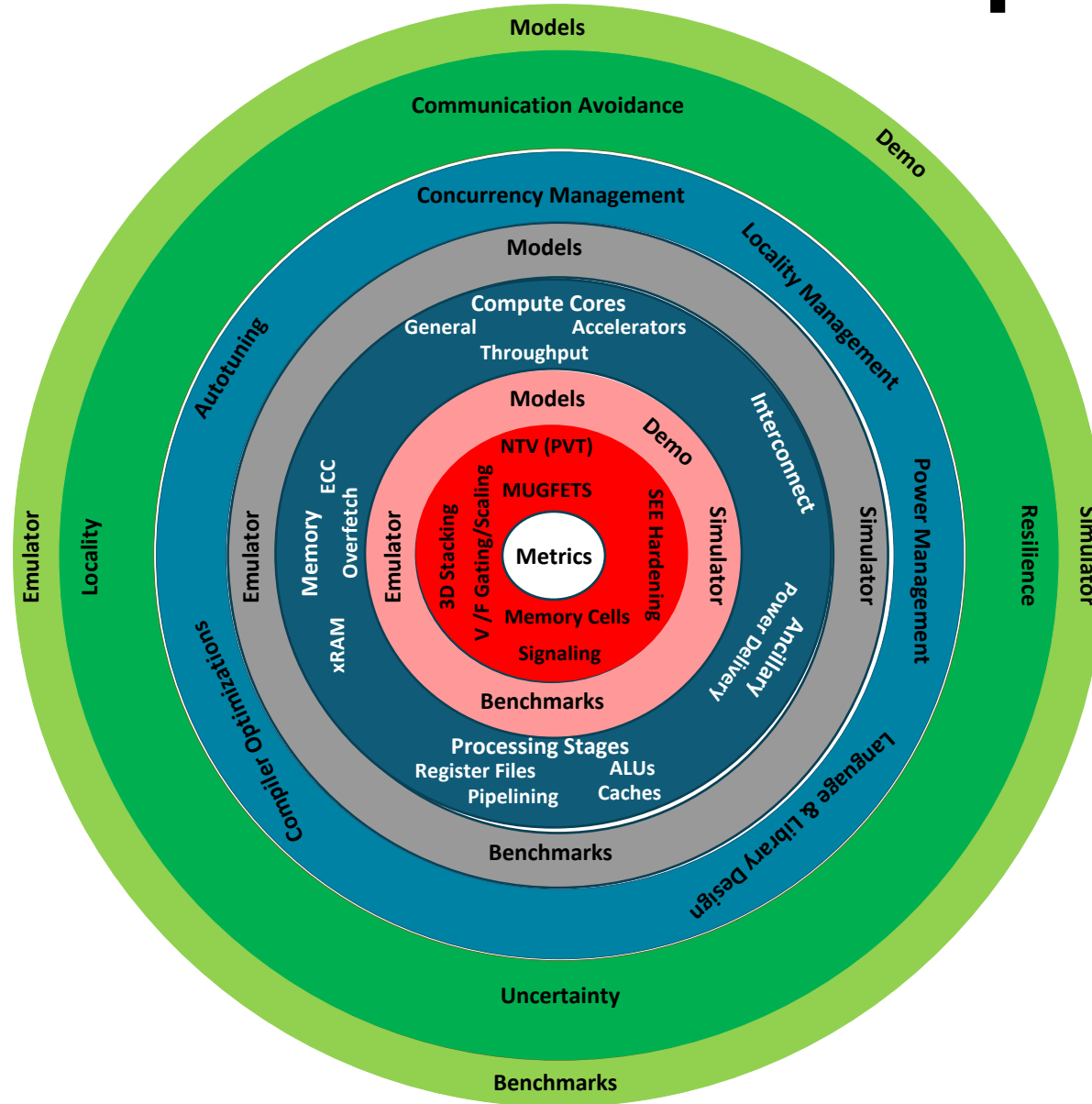
Codesign/ModSim in Action

-  Circuit Design
-  Circuit Modeling & Simulation
-  Architecture Design
-  Architecture Modeling & Simulation
-  System Software
-  Algorithm Design
-  Algorithm Modeling & Simulation



Pathway to Solutions is Complicated

- Circuit Design
- Circuit Modeling & Simulation
- Architecture Design
- Architecture Modeling & Simulation
- System Software
- Algorithm Design
- Algorithm Modeling & Simulation



The Vision: Ubiquitous Modeling

- Performance, Power and Reliability
 - Together!
- Bag-of-tools approach
 - Not one for all, but all for one.
 - Modeling, simulation, and emulation.
- Lifecycle coverage
 - HW-SW
 - From design space exploration to analysis of early implementation to deployment to runtime optimizations
- Co-design
 - Modeling needs to be applied to negotiate trade-offs at all boundaries of the HW-SW stack.
- Dynamic Modeling
 - Complexity of systems renders static/offline modeling insufficient
- Introspective Runtime
 - Dynamic HW-SW; rapid optimizations
 - Runtime system as model driven, and the model as actionable.

ModSim is the Quantitative Set of Codesign Methodologies

Essentially a multi-objective, non-convex optimization problem.

Bridges spectrum of scales, from devices/sensors to integrated systems to complex workflows

Need for dynamic modeling/codesign

ML-based capability for ModSim

- Developed for the complex workflows of experimental science
 - Extraordinary data challenges, combined with control, real time optimizations, introspection
- Dynamic simulations of complex environments comprised of socio and physical components
- Many technical challenges of DHS lend themselves to a dynamic codesign approach codesign – framework/foundation for digital twins

Challenges for System and Application Design

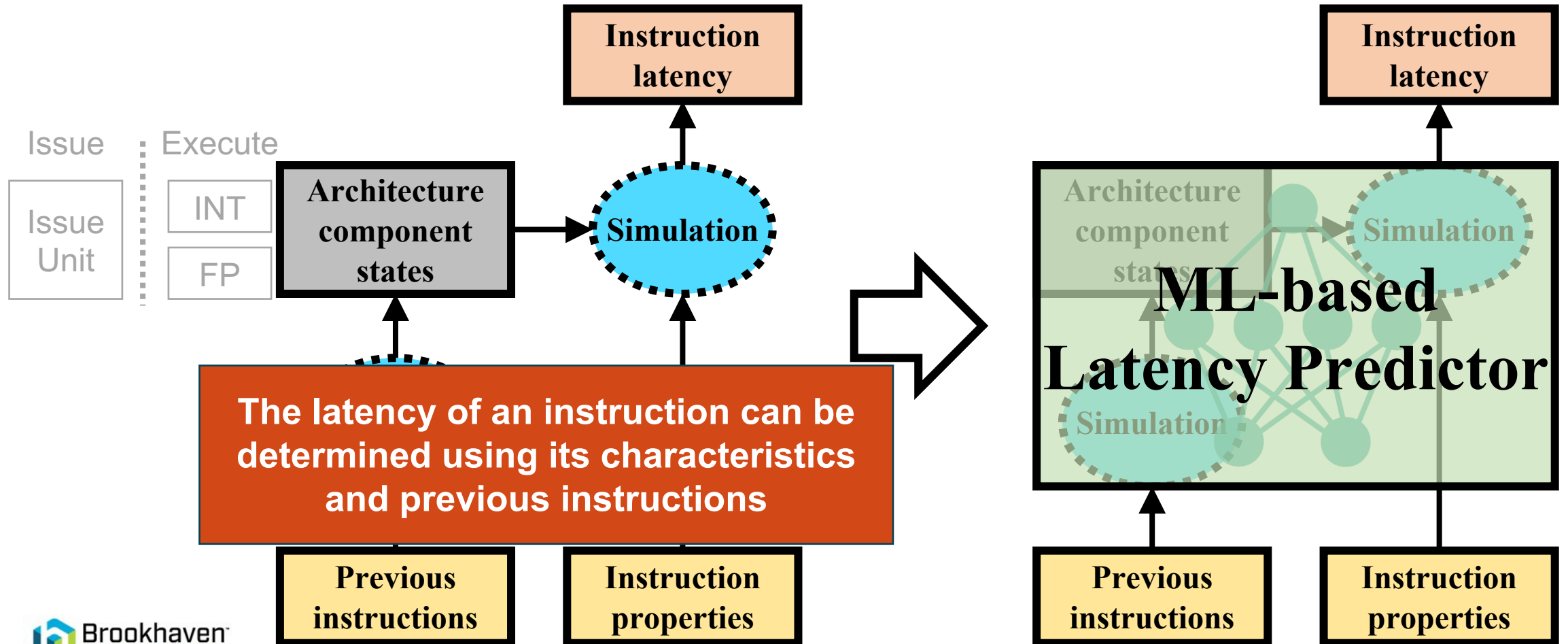
- Multiple constraints
 - Optimal performance
 - Power constraints
 - Reliability
- Adaptivity: vast numbers of “knobs” to deal with
 - Applications – data driven
 - Systems -- heterogeneous
- Complexity of the system software stack—dynamic behavior
 - Models in runtime
 - Actionable models
 - Guiding runtime optimizations and operation
- Complexity of the architecture and associated technologies
 - Need to leverage marketplace
 - Extreme-scale system are increasingly emerging as a synthesis of technologies
 - Leverage commoditization but adds specific smarts
- Modeling is called to capture multiple boundaries of the hardware-software (HW-SW) stack
- Applications must cope with and help mitigate the increased complexity
- Triggers the need for modeling now, wide-spread exploration of future applications and technologies

Why doing ML-based Simulation

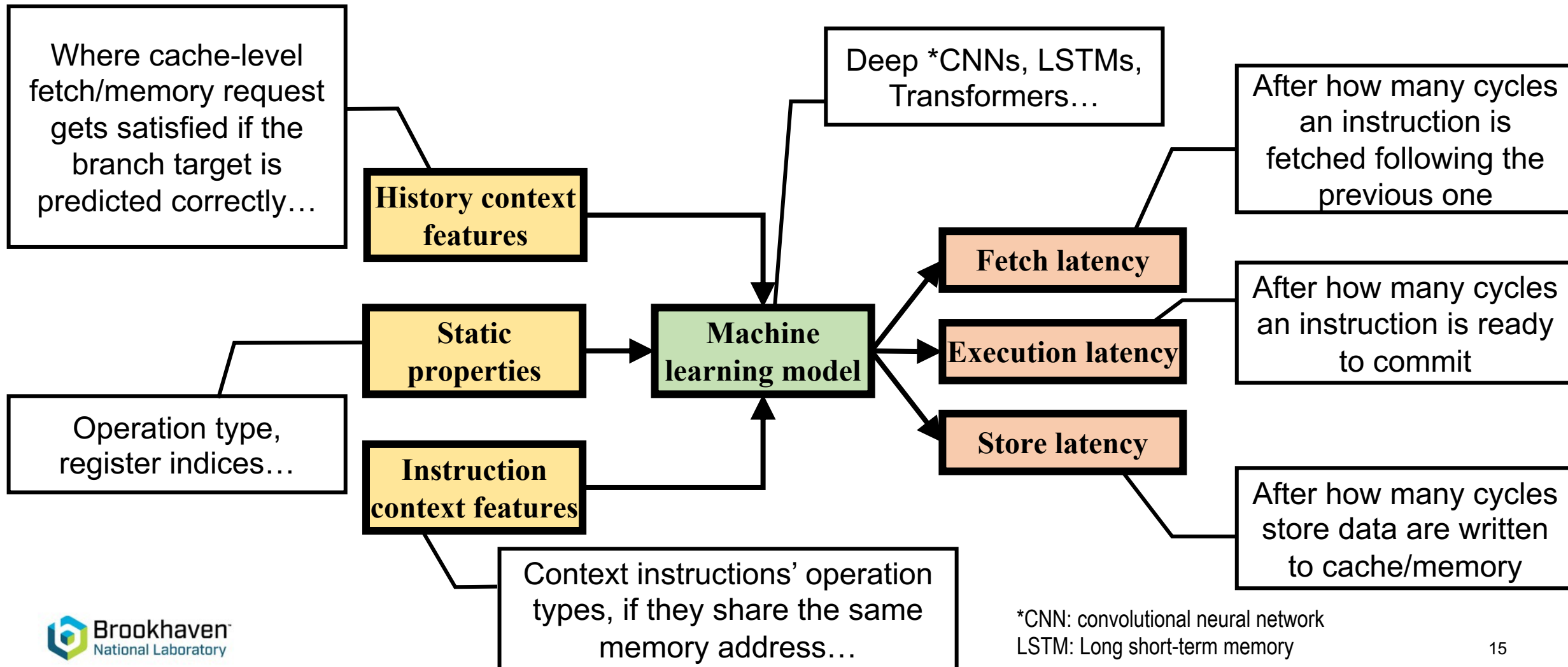
- ML models, especially deep neural networks, have been proven to be excellent function approximators
 - Can expect they can be applied to approximate the complex and implicit latency calculations that are essential to computer architecture simulation
 - ML is in widespread use from computer vision to scientific computing – a lot to learn from
- ML-based simulation is more flexible compared with ML-based analytical modeling because it does not require training per program/input.
- ML-based simulator could bring performance advantages –inference is highly parallel, and state-of-the-art accelerators and software infrastructures are well optimized for such tasks.

ML-based Simulation Foundation

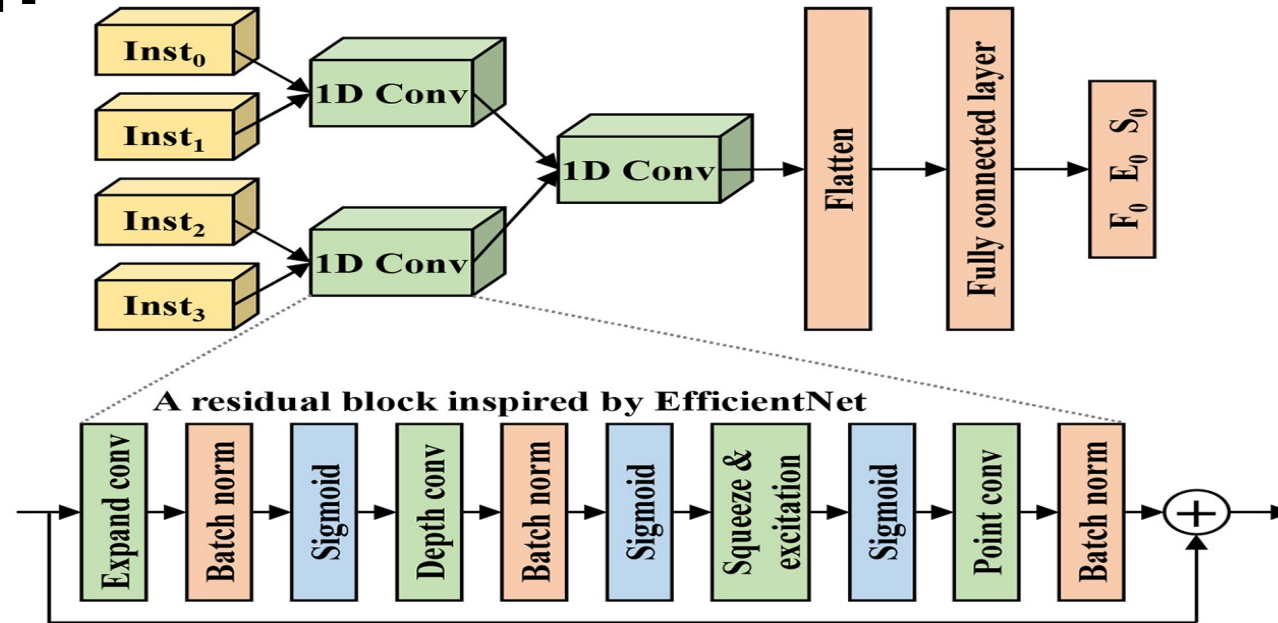
Goal: predict instruction latencies.



Instruction Latency Prediction



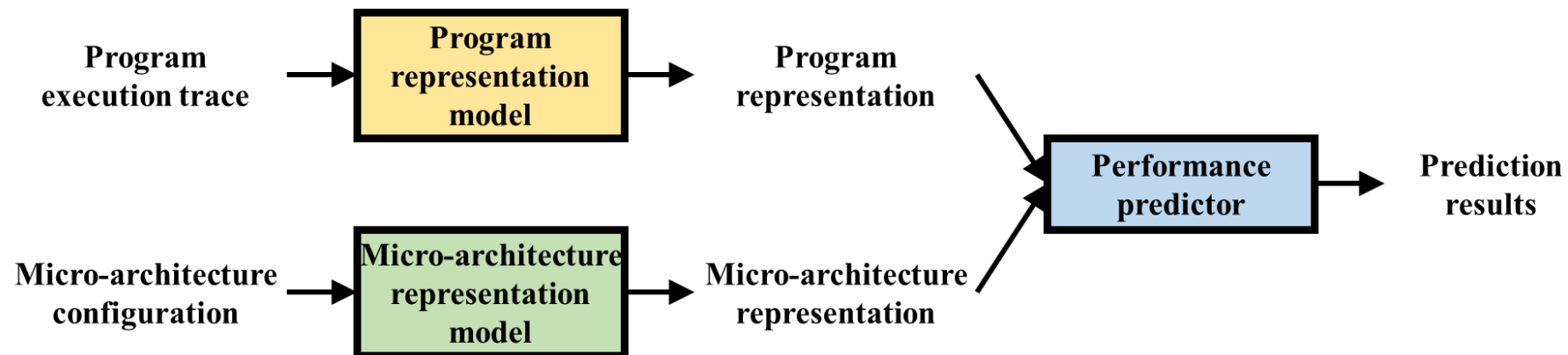
Specialized Neural Networks for Architecture Simulation



- We organize input instructions in a 1D array by their execution order -- features are channels
- We organize the convolutional layers in a hierarchical way
- A classification model could help to better distinguish between close latency values, where every latency value corresponds to a class,
 - ML model predicts which class has the largest probability.

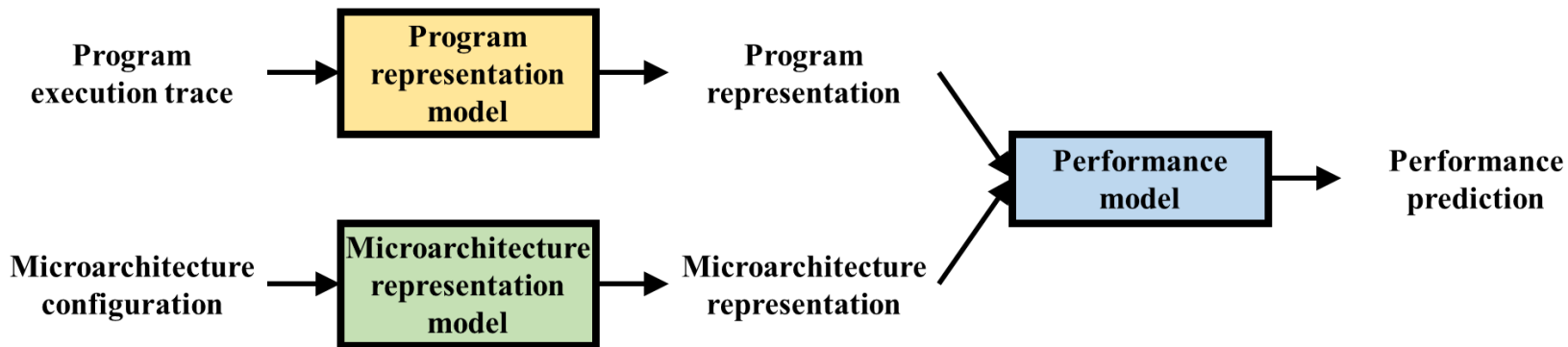
General Performance ModSim

- A general performance ModSim tool should separate the impact of program and microarchitecture
 - When one party changes, no need to re-model/simulate the other
- Basic idea: isolate the performance impact of program and microarchitecture using separate ML models
 - Program and microarchitecture representations are general and reusable.



Representation-based Performance Modeling

Basic idea: separate the impact of program and microarchitecture in the ML model architecture; learn general and reusable program and microarchitecture representations.



Program features need to be microarchitecture-independent:

- Convention features, such as operation types; register indices
- To capture cache performance: use reuse-distance as features
- To capture branch prediction performance: use branch entropy as features

Design Space Use Case Scenario

Representation-based modeling is broadly applicable in many use cases.

Example: L1 and L2 cache size exploration to minimize the objective function

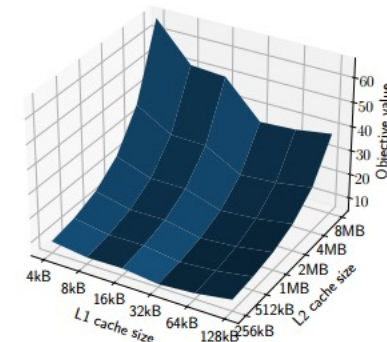
- Objective function: $\text{execution_time} * (1000 + 10 * \text{L1_size} + \text{L2_size})$
- Select the best cache sizes for 17 SPEC benchmarks

Time overhead to finish this design space exploration

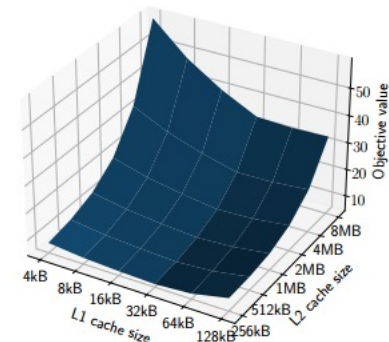
- Our method: 5 hours of gem5 simulation + 6 hours of model training
- gem5 simulation: 600 hours
- Previous ML-based performance models: 200 hours of gem5 simulation

Objective function values under various cache sizes

- Our method selects the top 3.6% designs on average



(a) gem5.



(b) PerfVec.

Dynamic Codesign

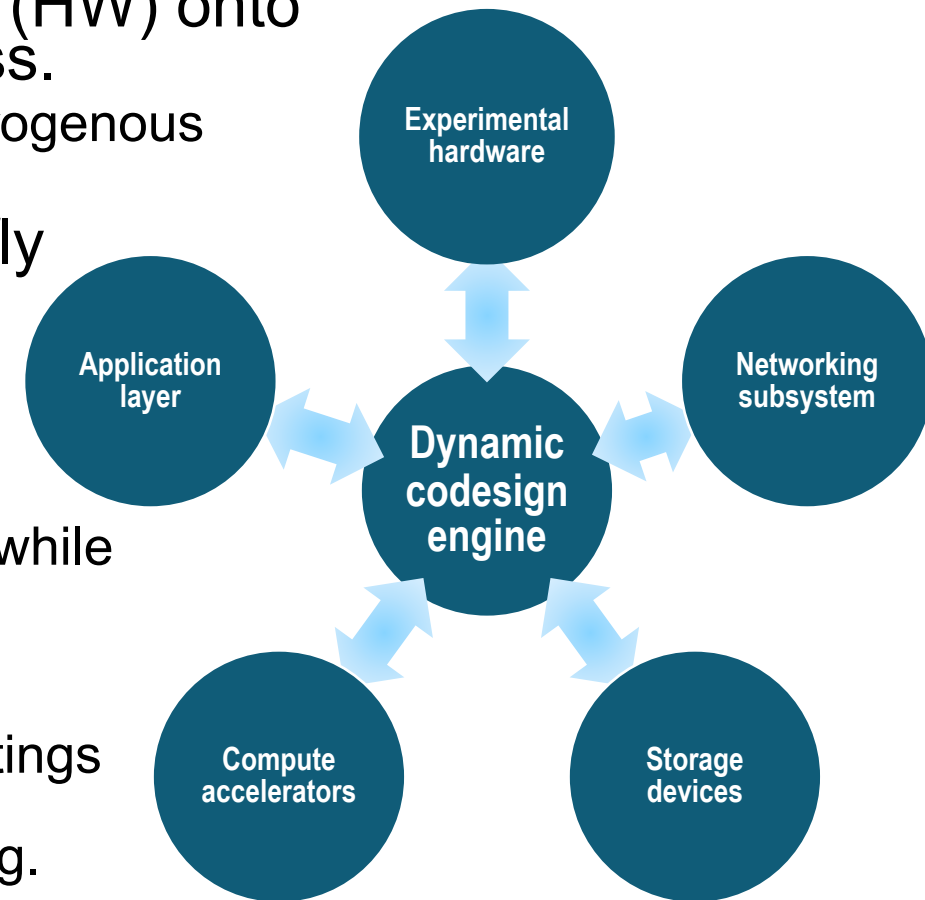
Codesign is not merely static mapping of hardware (HW) onto software (SW), but a dynamic, data-driven process.

- Vital for now dominant data-driven workloads and heterogenous architectures

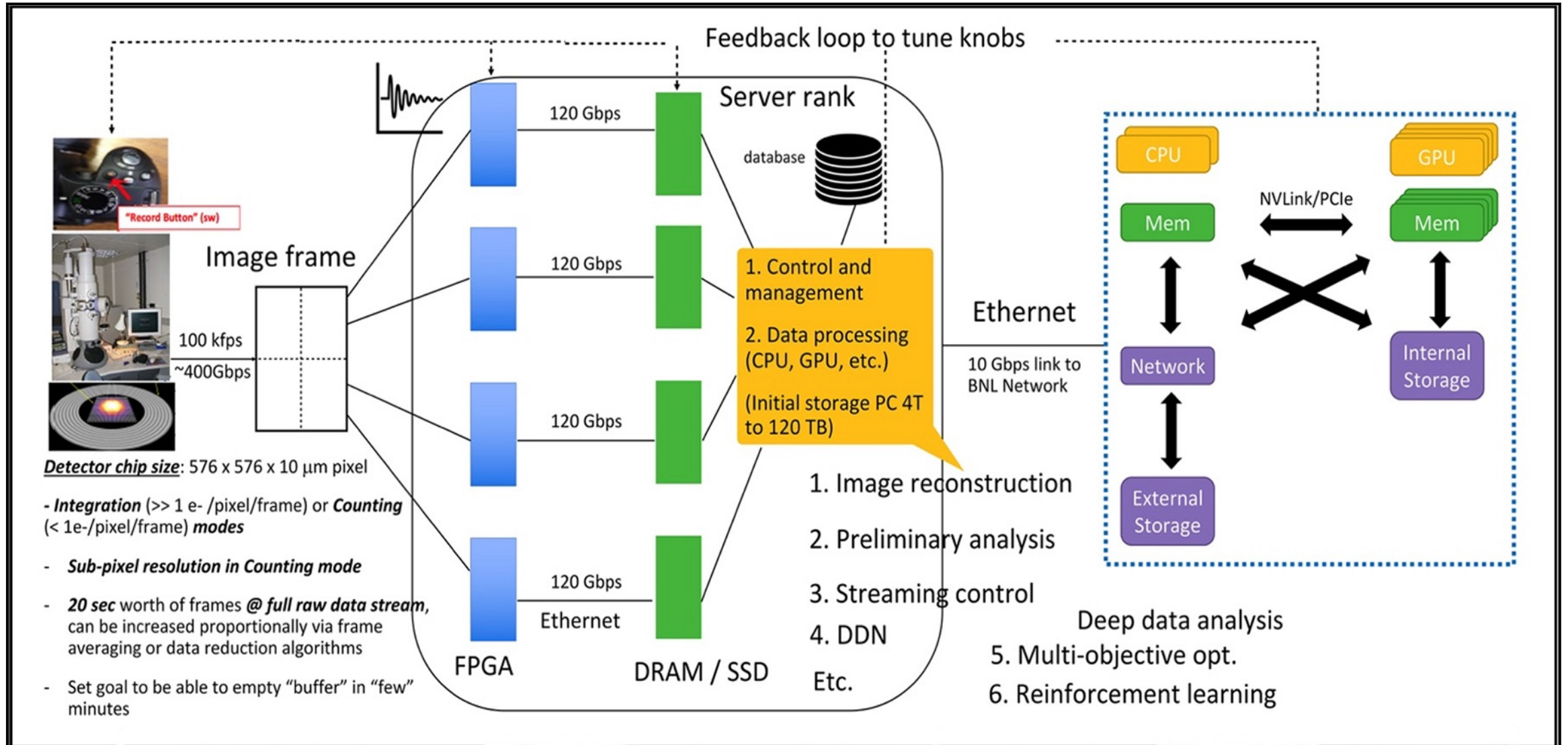
Ability to find new placements or mappings on the fly is needed.

In this regime, experimental HW, application SW, and other devices (e.g., storage) are all coupled through feedback loops.

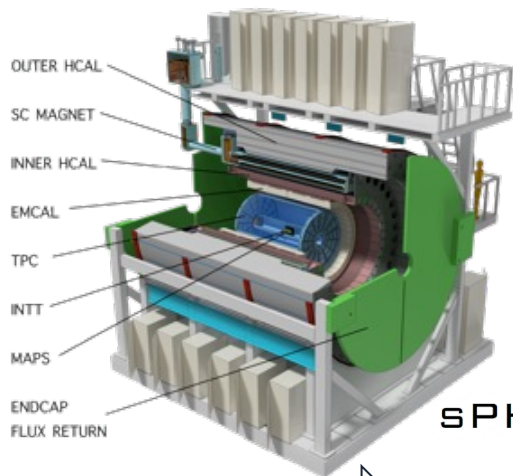
- Rational, quantitative ways to reconfigure components while experiments are conducted.
- Gather training data from actual experimental and SW configurations.
- Use ML models to predict optimal actions and knob settings for a dynamic codesign engine.
- Train the intelligent runtime using reinforcement learning.



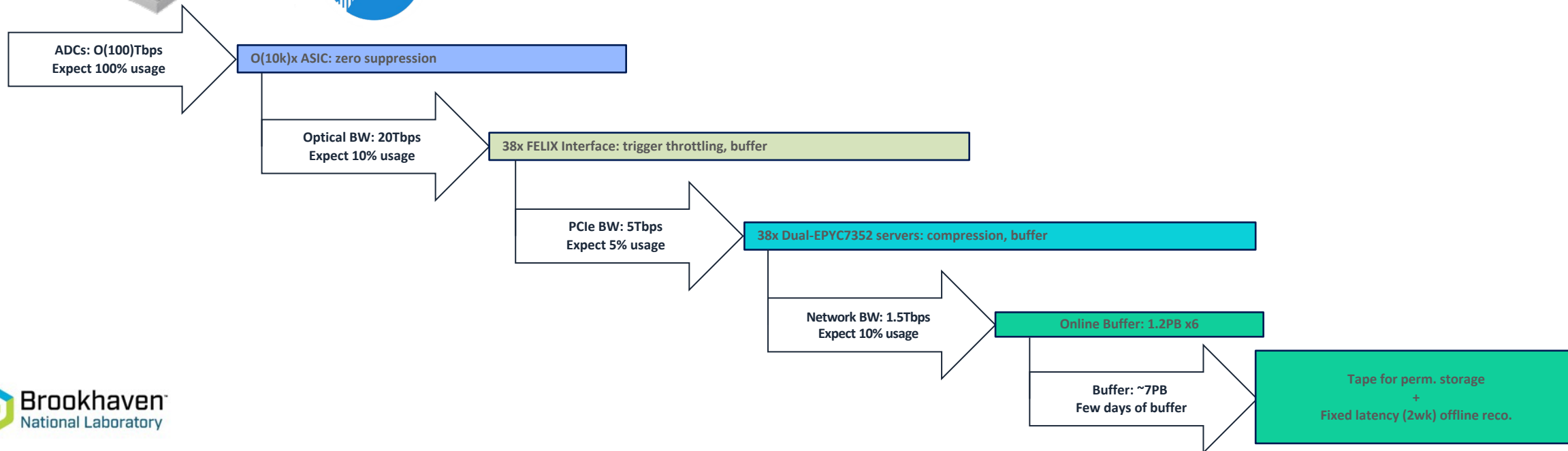
Workflows du Jour: Microscopy, ...

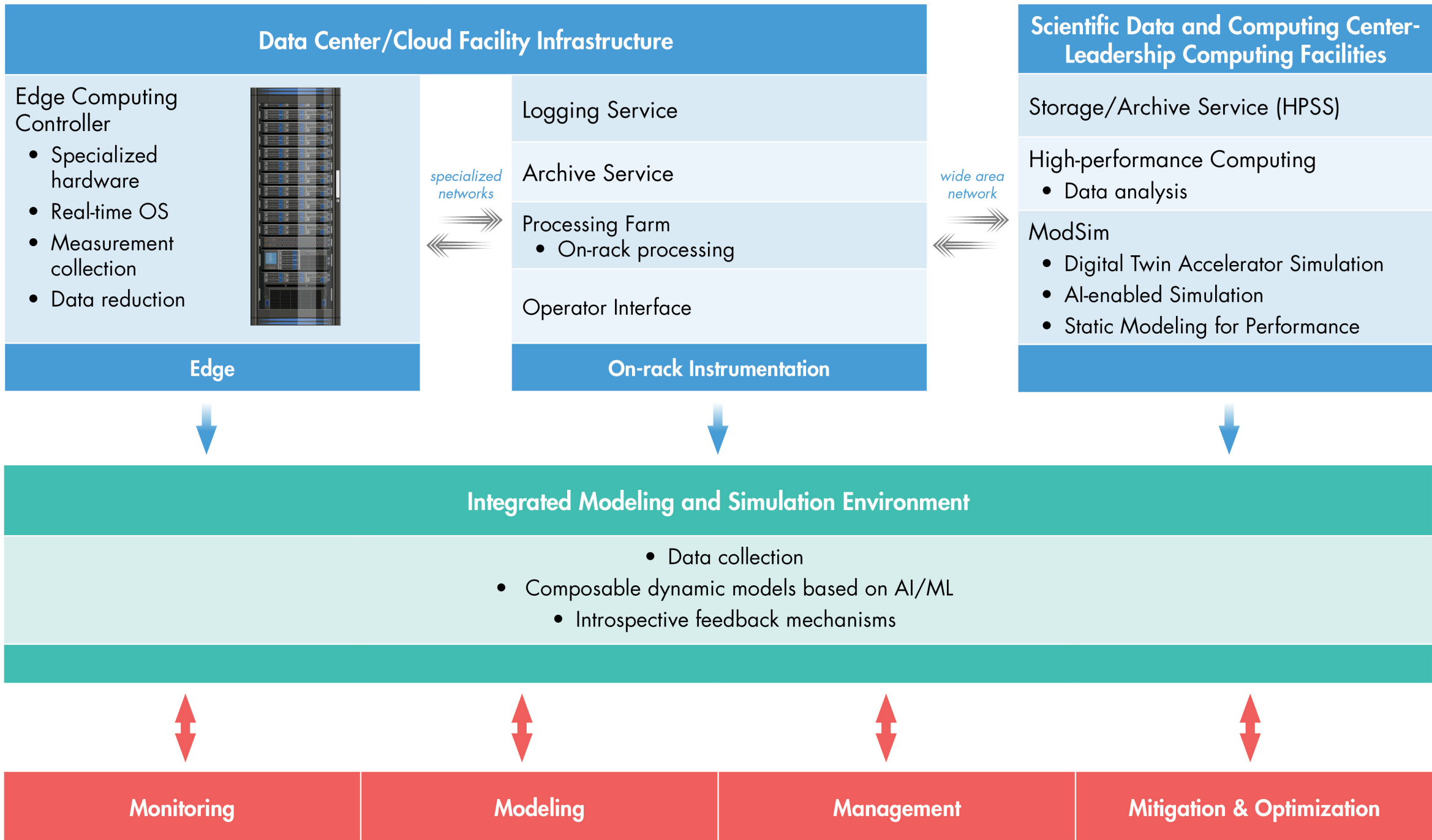


Accelerator Detector Workflow



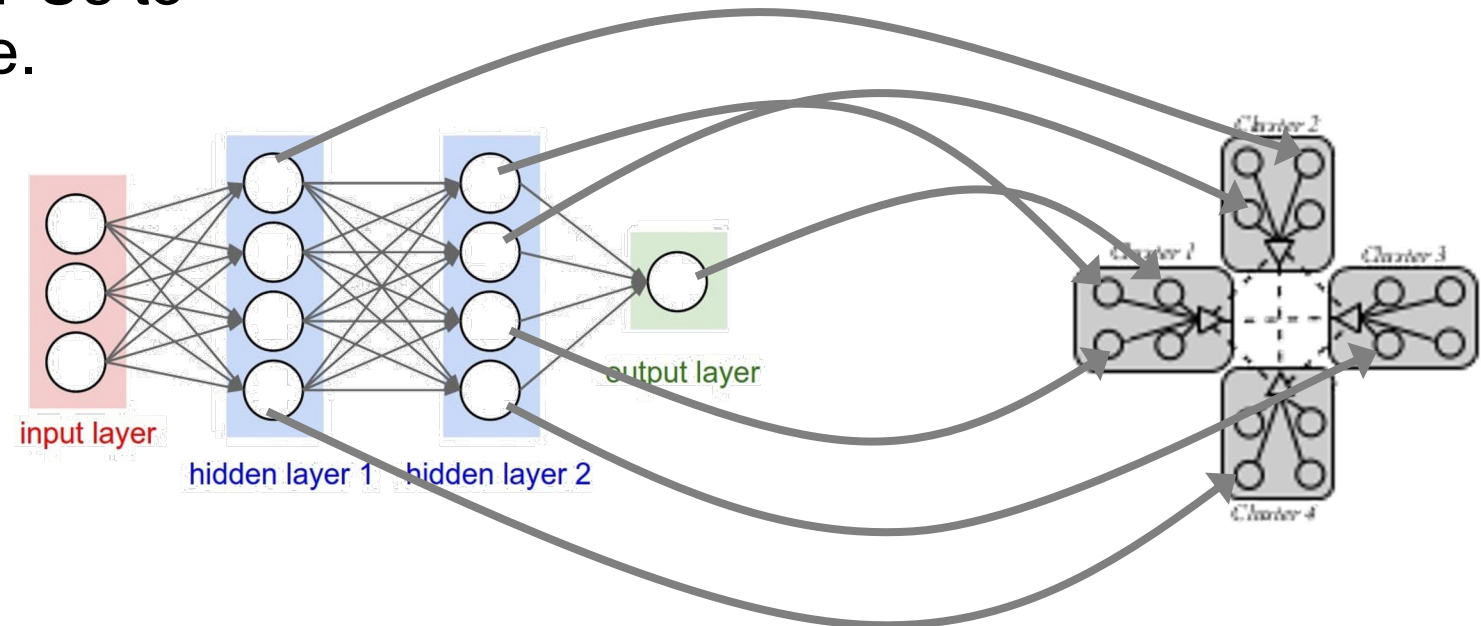
- Next-generation collider experiment: sPHENIX starts data-taking in 2023
- Streaming DAQ at 2Tbps readout also applicable to future electron-ion collider (EIC)
- Opportunities for real-time high-throughput noise-rejection, anomaly detection monitoring, and reconstruction





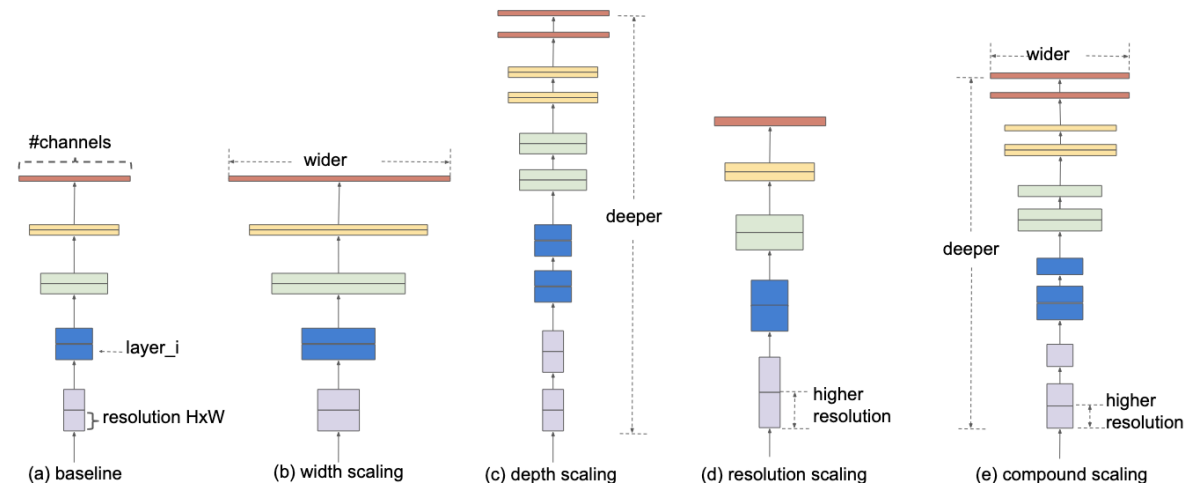
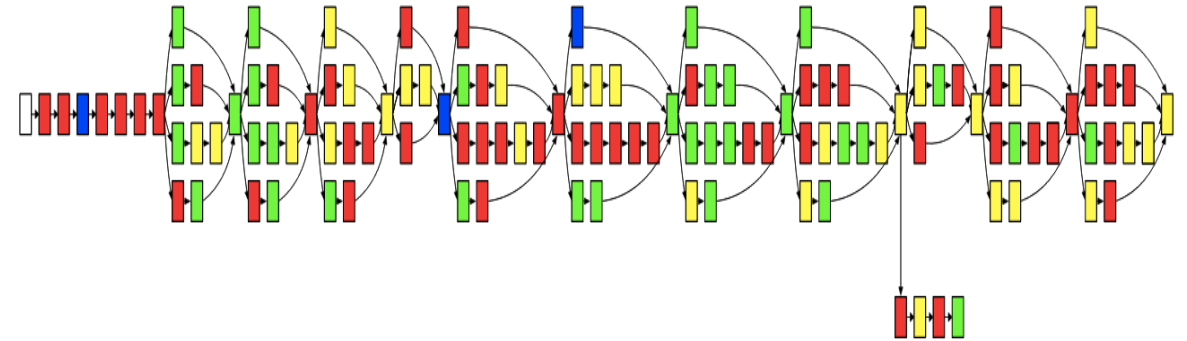
Neural Nets for Device Placement

- How to automatically map a numerical algorithm onto available hardware resources?
 - Example: In machine learning, map a deep neural network onto a system of GPUs/CPUs to optimize training time.
- Combinatorial optimization problem
 - N operations, D devices $\rightarrow D^N$ possible mappings.



Beyond Static Codesign Approaches

- Promising work on optimizing ML workloads
 - Device placement: how to place elements of a computation graph onto available accelerator cores
 - Resizing neural nets: How to trade off accuracy for model size.
- However, dynamic approaches are needed:
 - Different parts of an experiment call for disparate imaging settings (impacting data resolution/rate)
 - Algorithm settings change (e.g., required accuracy)
 - Shifting demands may require different HW-SW mappings for optimal performance



Top: Device Placement Optimization with Reinforcement Learning. Mirhoseini et al. (2017)
Bottom: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Tan and Le (2019)

REDWOOD: ML-based Optimization of Complex Experimental Science Workflows for Distributed Resilience

Overarching goals:

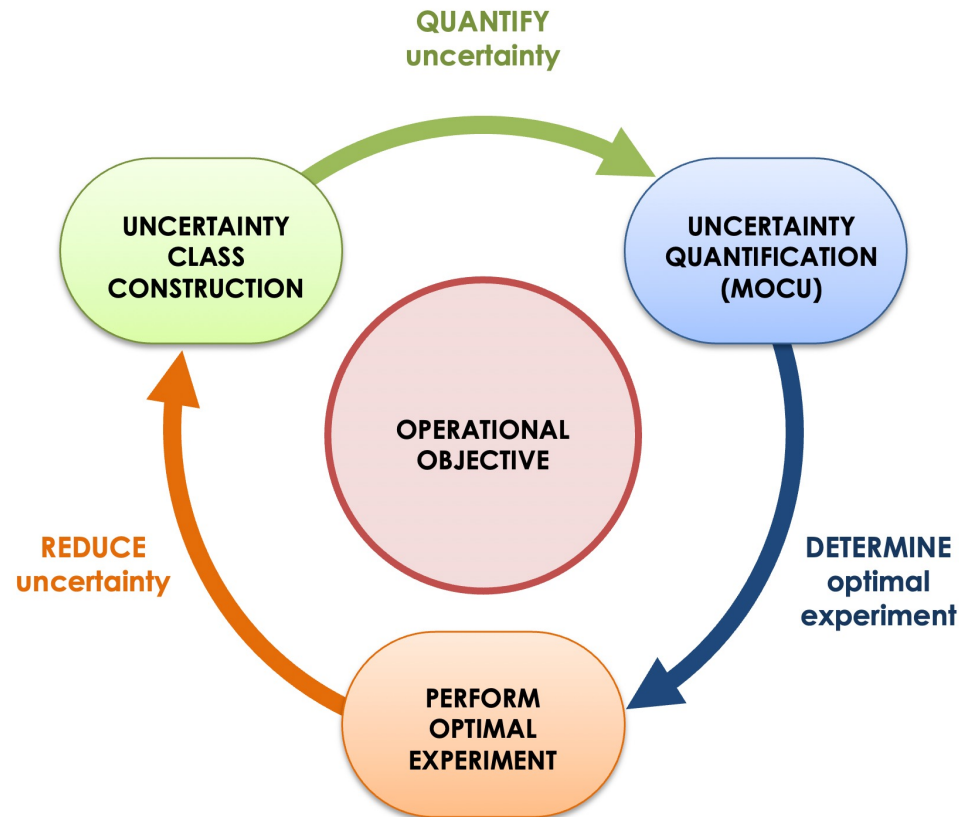
- Dynamic modeling (these methods can quantitatively capture dynamic and adaptive workflow and system behavior at runtime)
- Complex high-throughput (HT) and NRT workflows resilience
- Optimal data placement and resources utilization

ModSim specific:

- Optimize the resilience of the workflows by intelligently placing the data and processing across the distributed resources.
- Develop an intelligent, introspective, and dynamic workflow by drawing on a system model based on years of data captured from a large-scale production system. It will offer control and management capabilities required at all timescales—from NRT to delivery of scientific insight.
- Flexible resource provisioning
- Dynamic resource management
- Multi-dimensional scheduling optimization

Optimal Experimental Design (OED)

Fundamentals of an OED framework



- **Models and experiments are inherently uncertain:** theoretical approximations, experimental errors, imperfect information, etc.
- **Experiments are pointwise, costly, and time consuming**
- **Design experiments systematically to yield the highest amount of insight** from each experiment
- Make design decisions under uncertainty
- Demonstrated in drug discovery, superconducting quantum circuits, etc.

Summary

ML-based methods and tools for performance, power, and reliability hold great promise to advance the state-of-the-art in ModSim

Offer a flexible framework for codesign of complex (socio-)technical systems

Have direct applicability to challenges in the science and other mission spaces

Quantitative tool of codesign – framework for digital twins

From characterization to design to actionable capabilities