Integrated Hardware Architecture and Device Placement Search

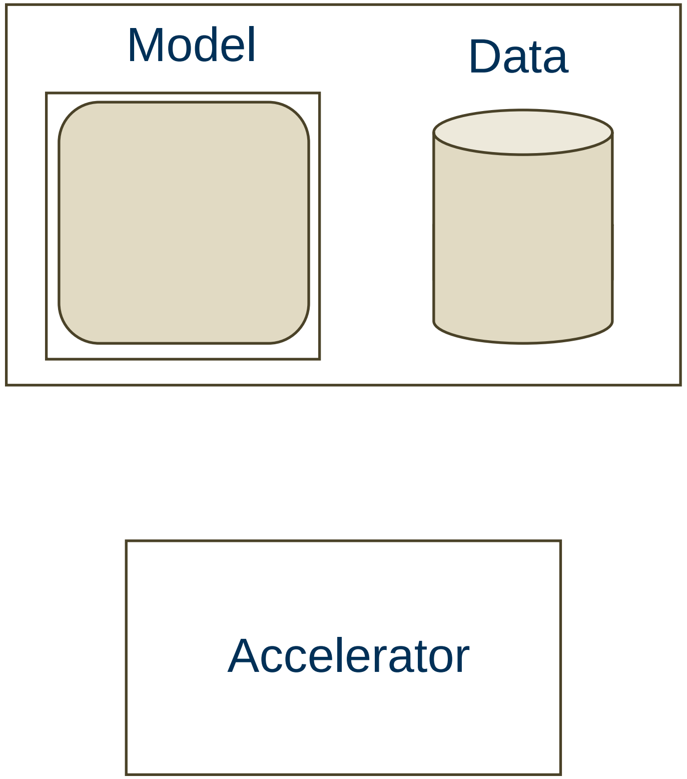
Irene Wang[1], Jakub Tarnawaski[2], Amar Phanishayee[2], Divya Mahajan[1]

[1] Georgia Institute of Technology, [2] Microsoft Research

Georgia Tech

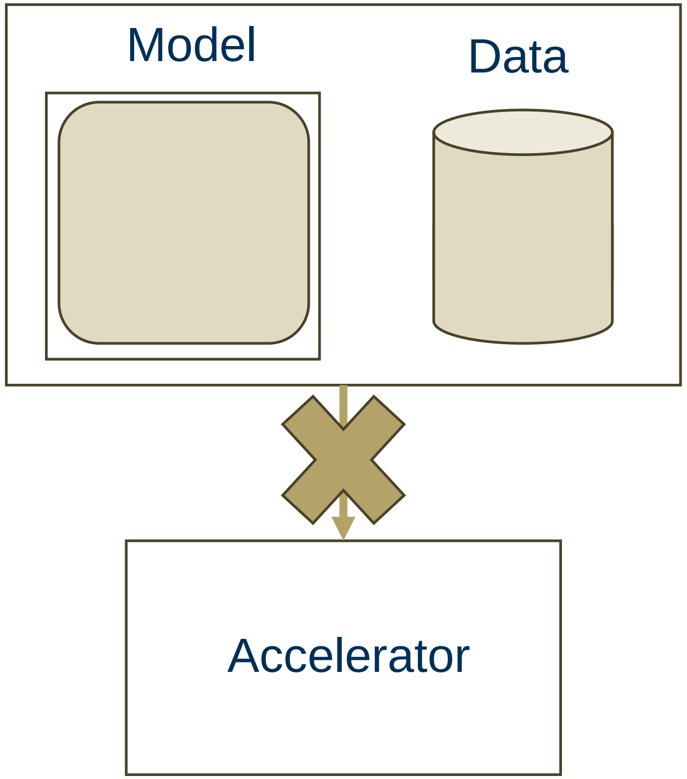
How to train DNN efficiently ?

State-of-the-art models are too large to fit on a single accelerator and need to be trained in a distributed manner



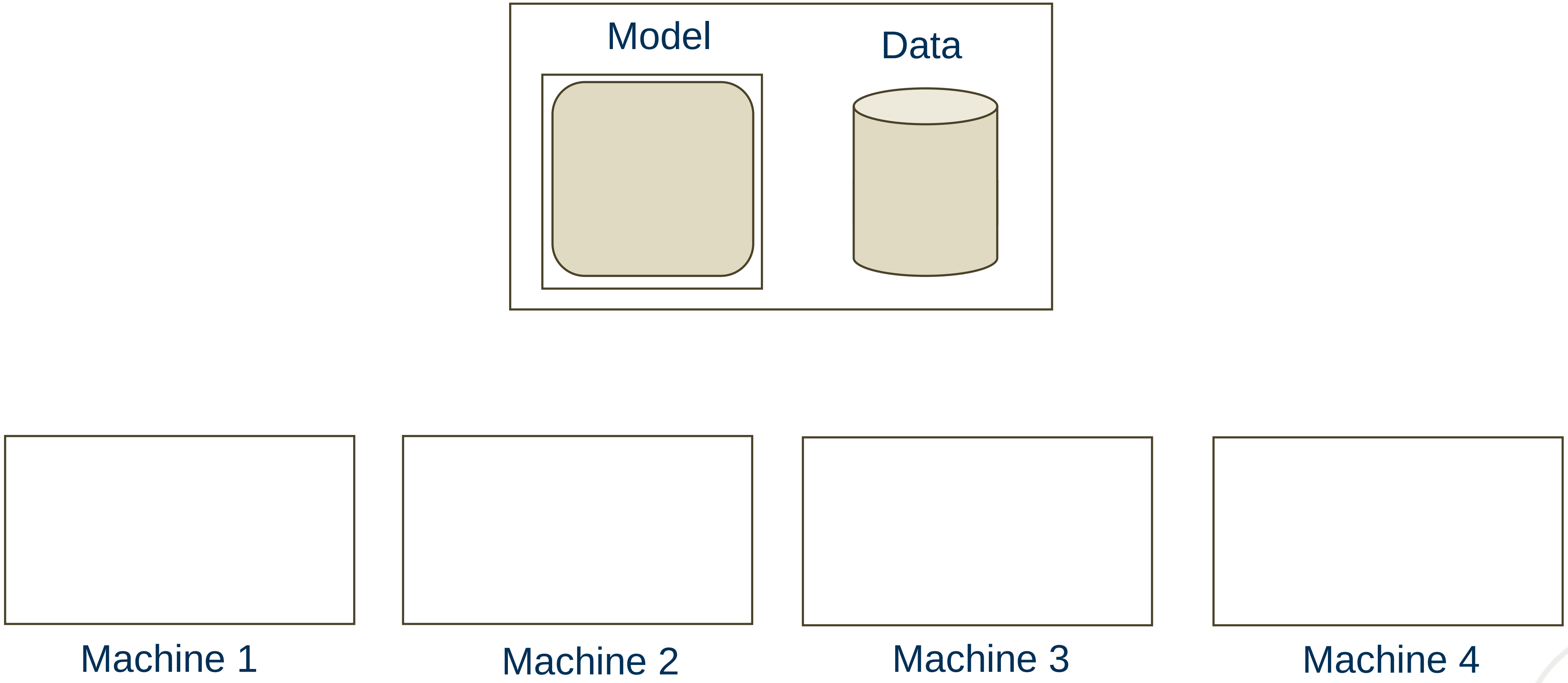
How to train DNN efficiently ?

State-of-the-art models are too large to fit on a single accelerator and need to be trained in a distributed manner



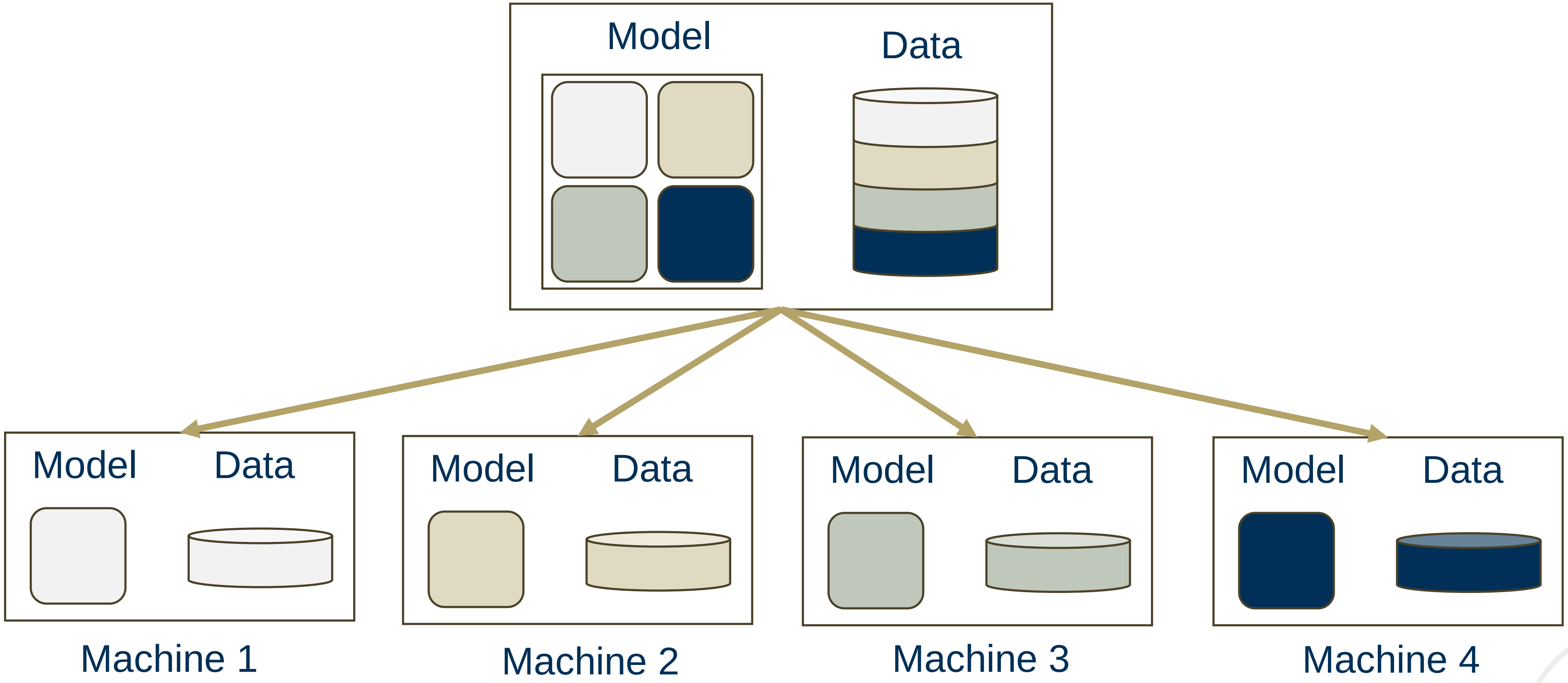
How to train DNN efficiently ?

State-of-the-art models are too large to fit on a single accelerator and need to be trained in a distributed manner



How to train DNN efficiently ?

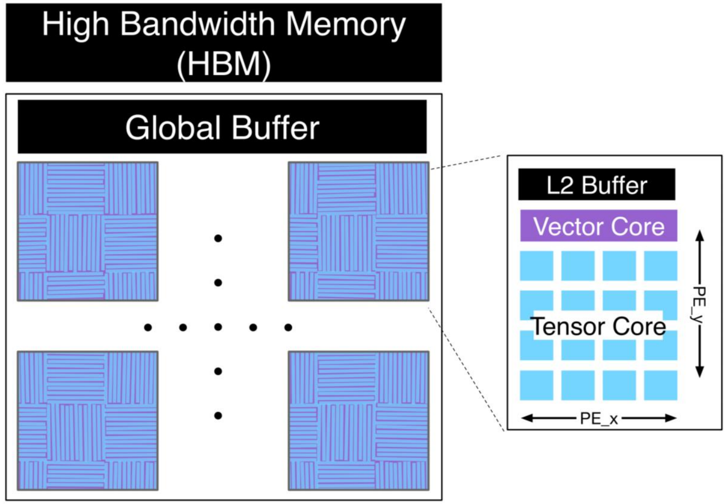
State-of-the-art models are too large to fit on a single accelerator and need to be trained in a distributed manner



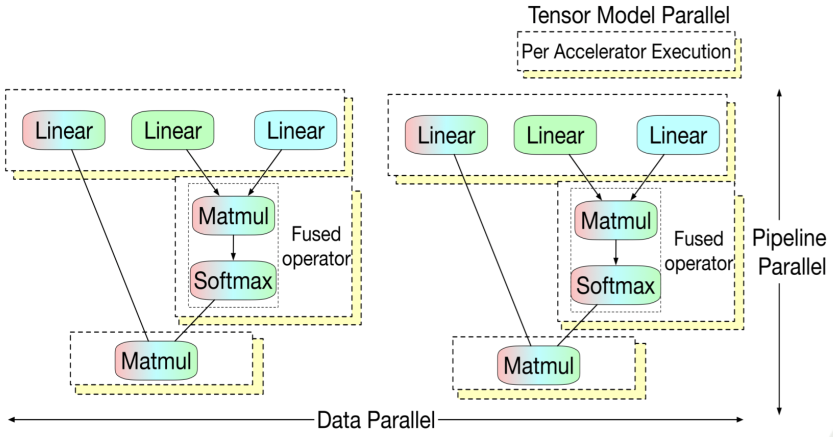
How to train DNN efficiently ?

Training DNN models require **2 simultaneous design choices** to be made to balance resource utilization and memory footprint

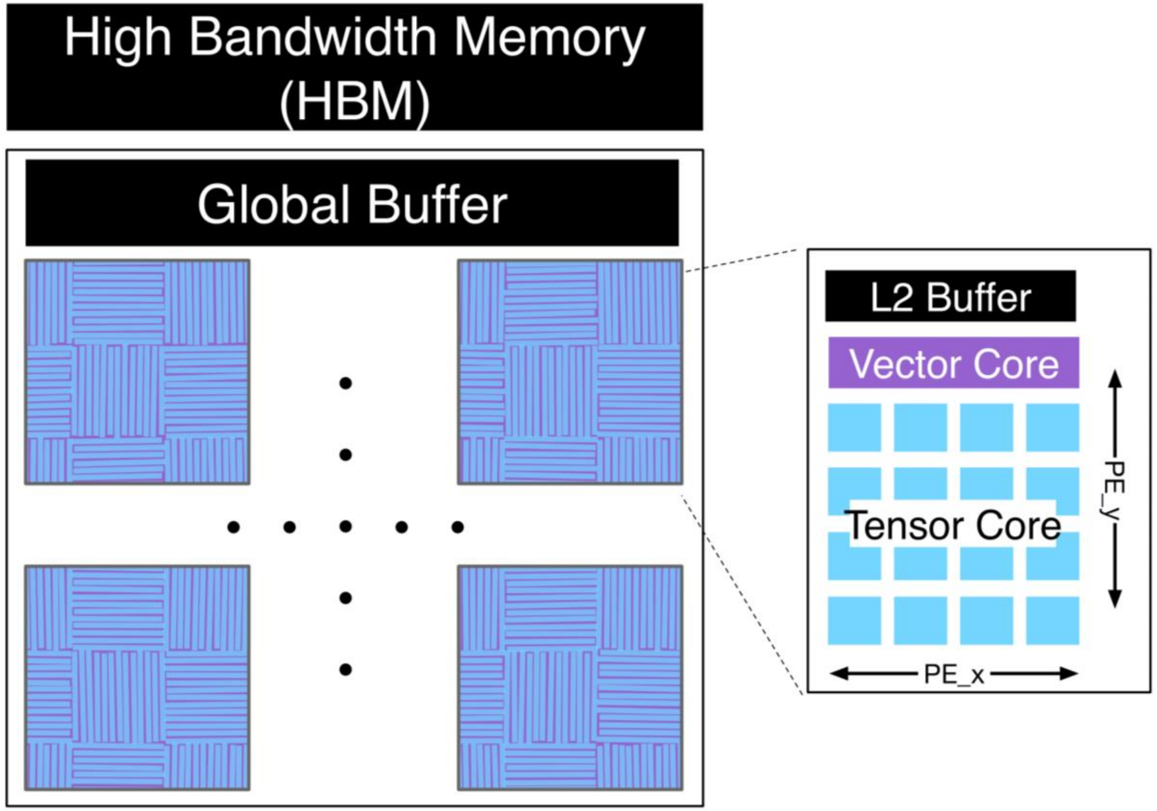
1. Hardware Architecture



2. Device Placement Strategy

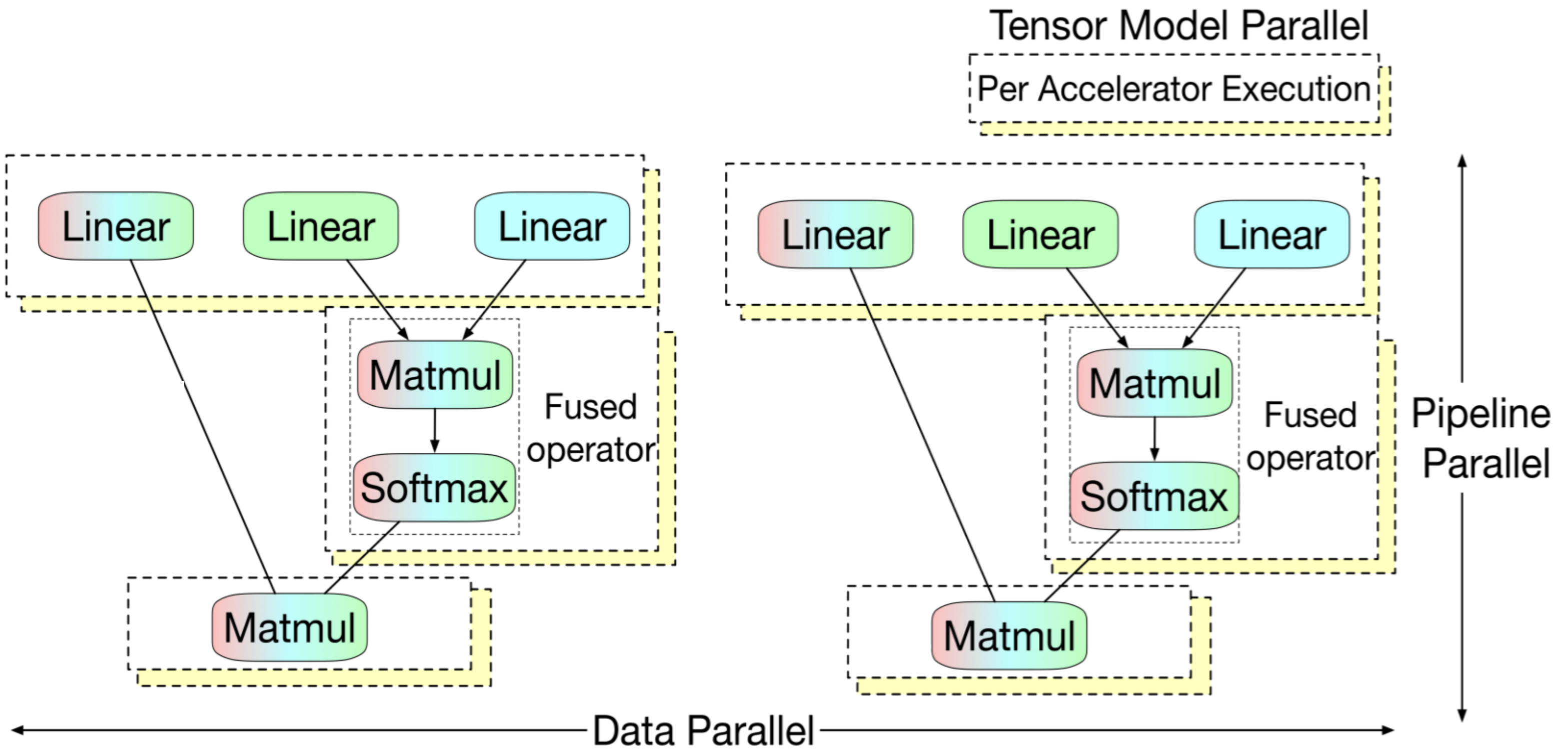


Design Choice 1: Hardware Architecture



Explores the on-chip and off-chip resource utilization

Design Choice 2: Device placement



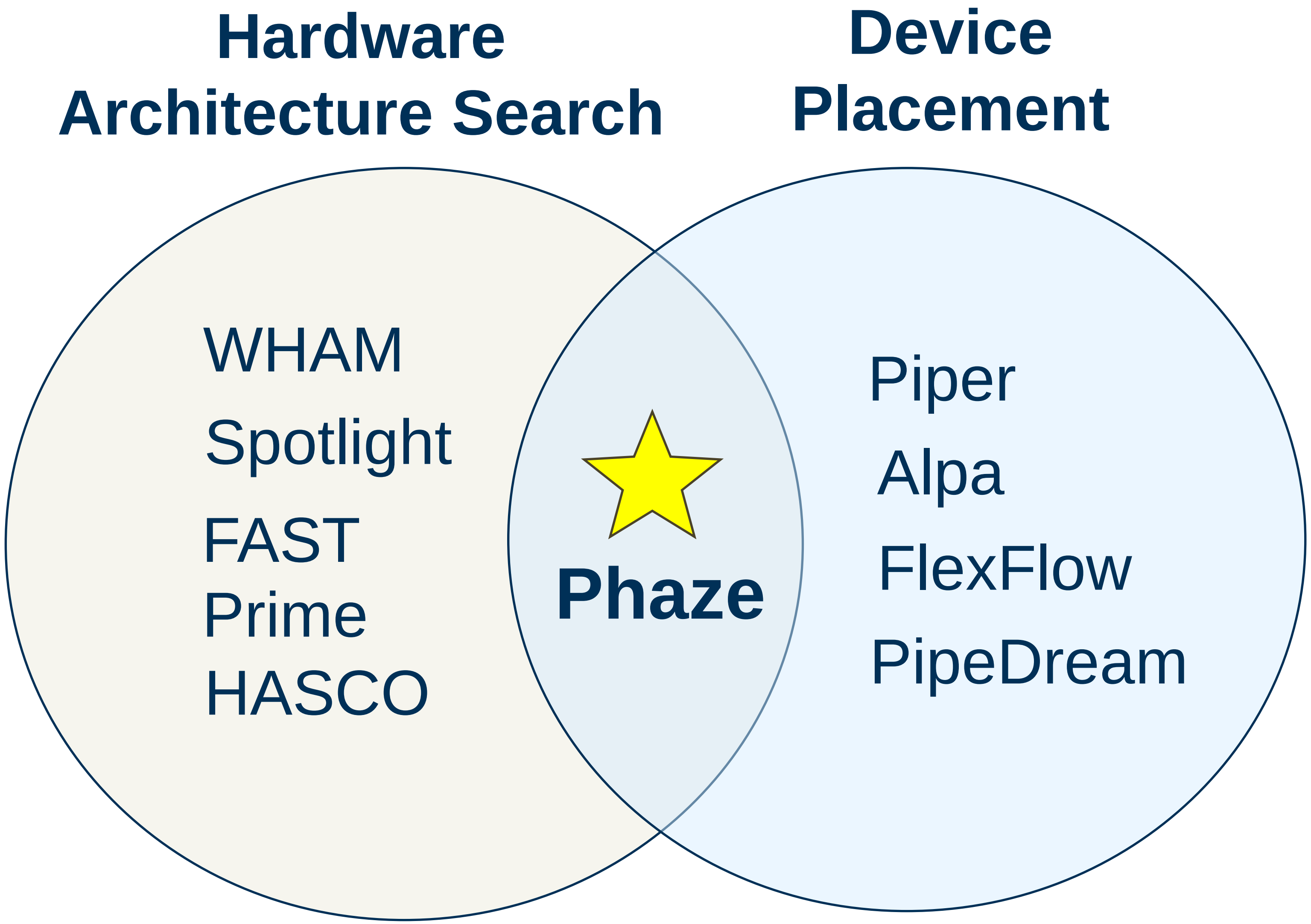
Balance between:

- Memory footprint
- Networking overhead
- Overall training throughput

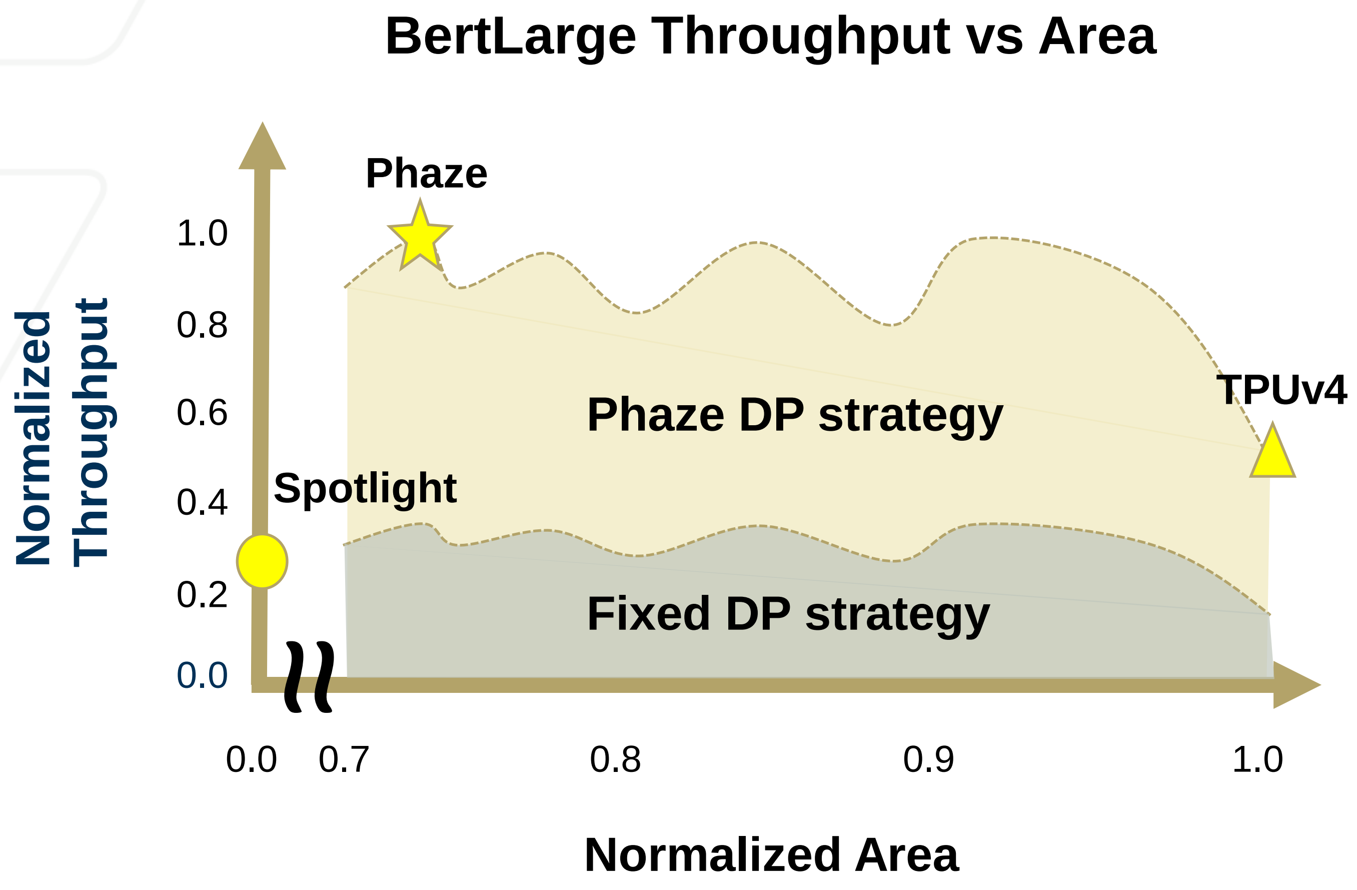
Motivation

What **architecture** and **model distribution strategy** can achieve the optimal performance for end-to-end deep learning training?

Prior Works


Hardware
Architecture Search
Device
Placement
WHAM
Spotlight
FAST
Prime
HASCO
Phaze
Piper
Alpa
FlexFlow
PipeDream

Need for Co-optimization



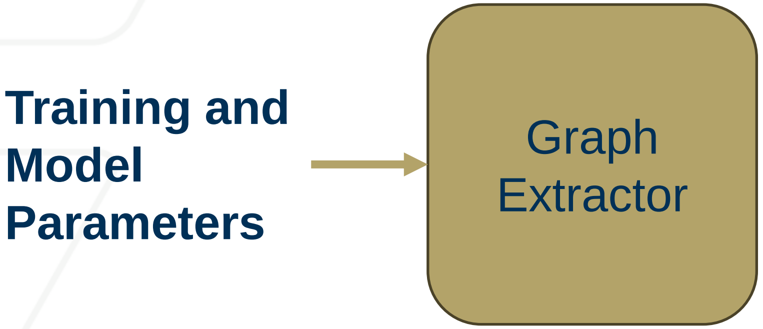
Fixed device placement in architecture search may lead to hardware under-utilization

Fixed hardware architecture in device placement search limit the search space of memory footprint and networking overhead

Phaze

Framework for co-optimizing **hardware architecture**, **device placement** strategy and per-chip **operator scheduling**

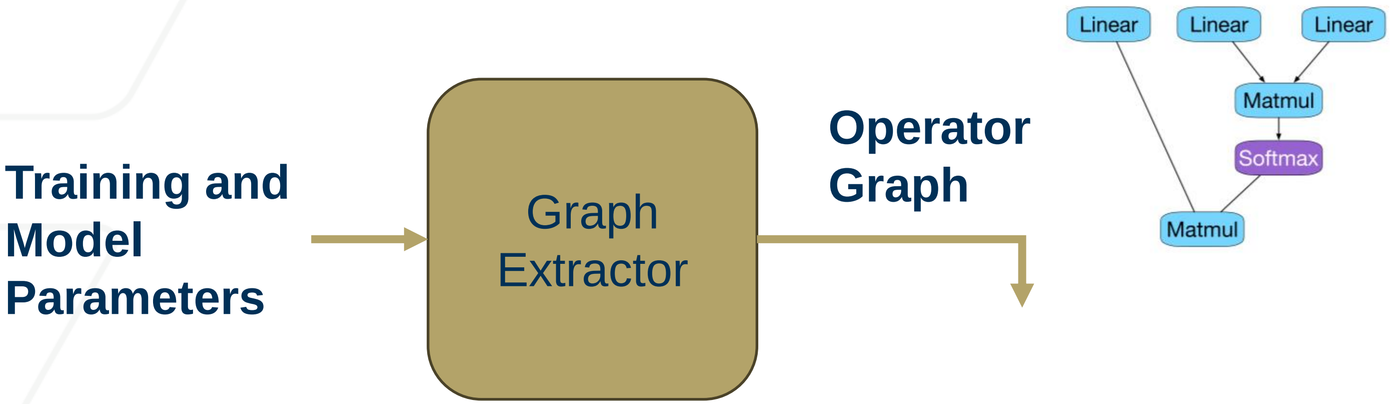
Overview of Phaze



Training and Model Parameters

Graph Extractor

Overview of Phaze

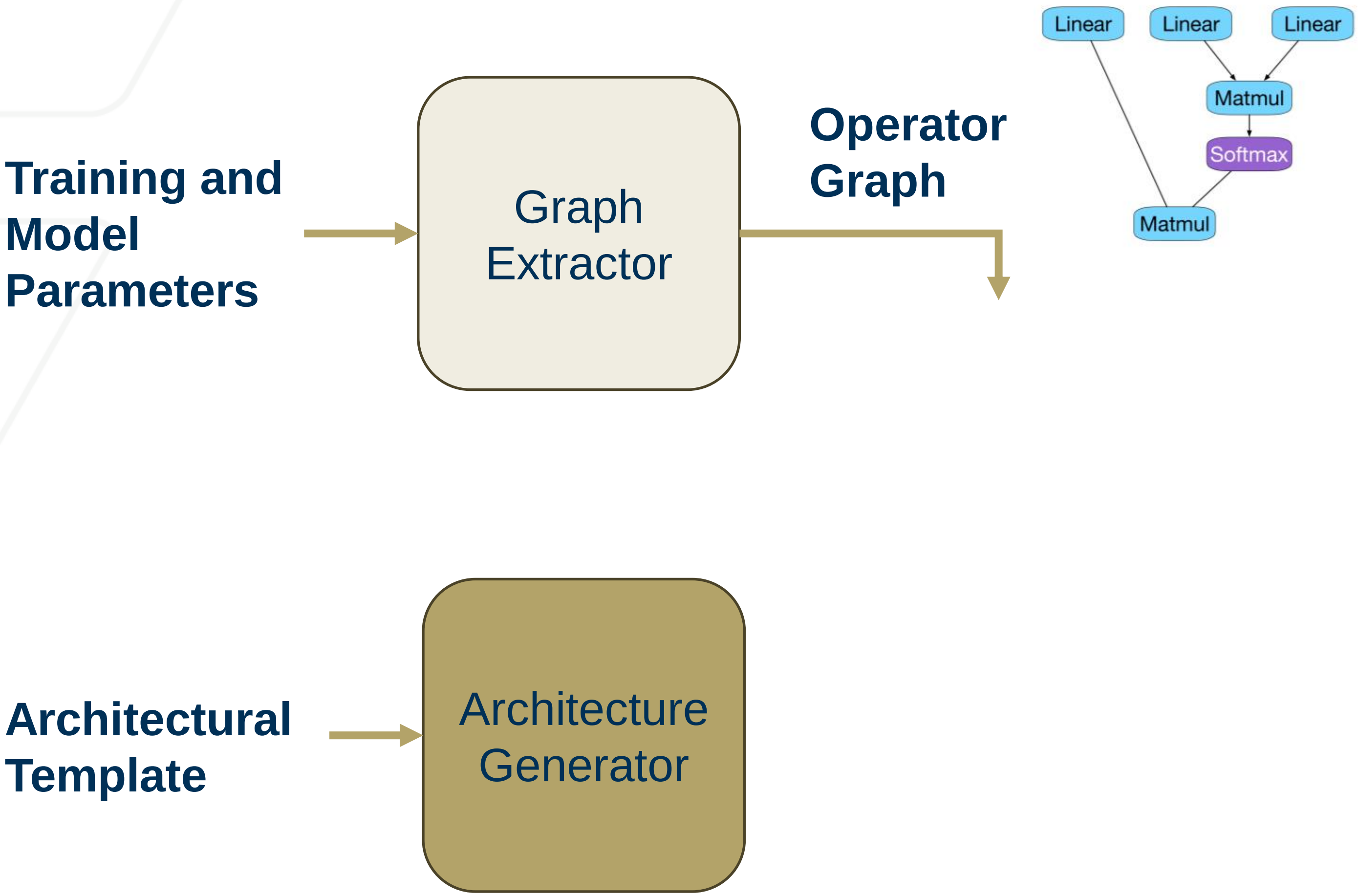


Training and Model Parameters

Graph Extractor

Operator Graph

Overview of Phaze



Training and Model Parameters

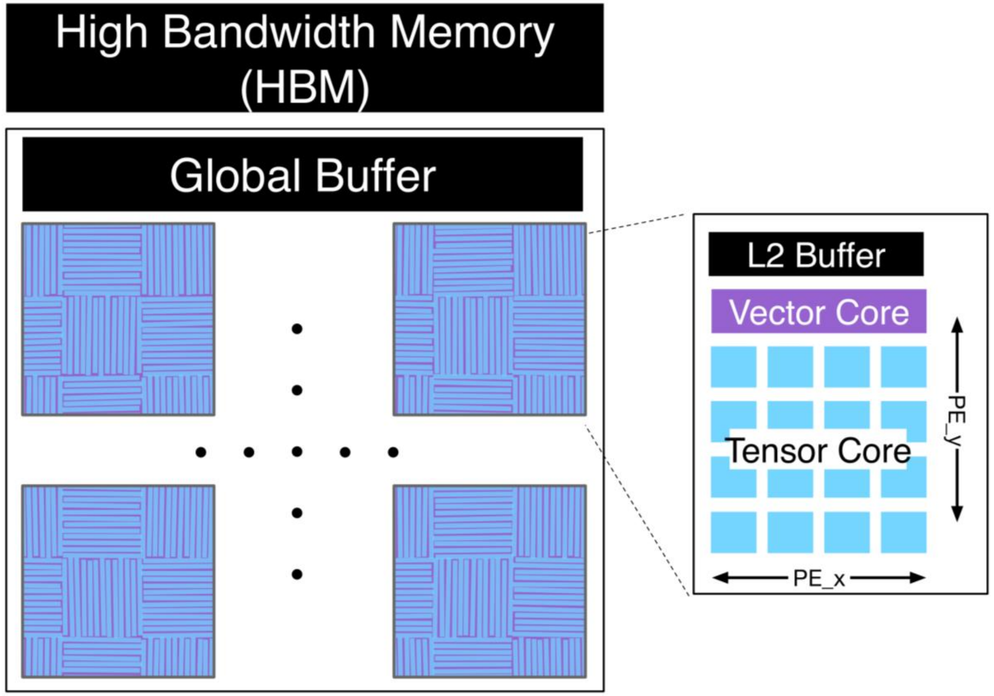
Graph Extractor

Operator Graph

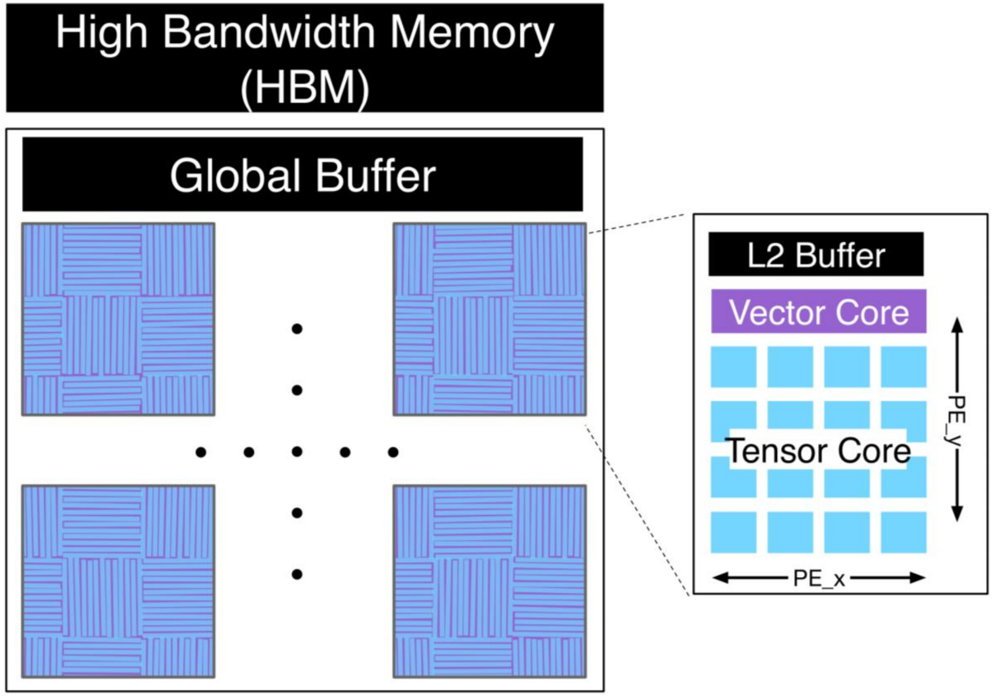
Architectural Template

Architecture Generator

Overview of Phaze: Architecture Generator



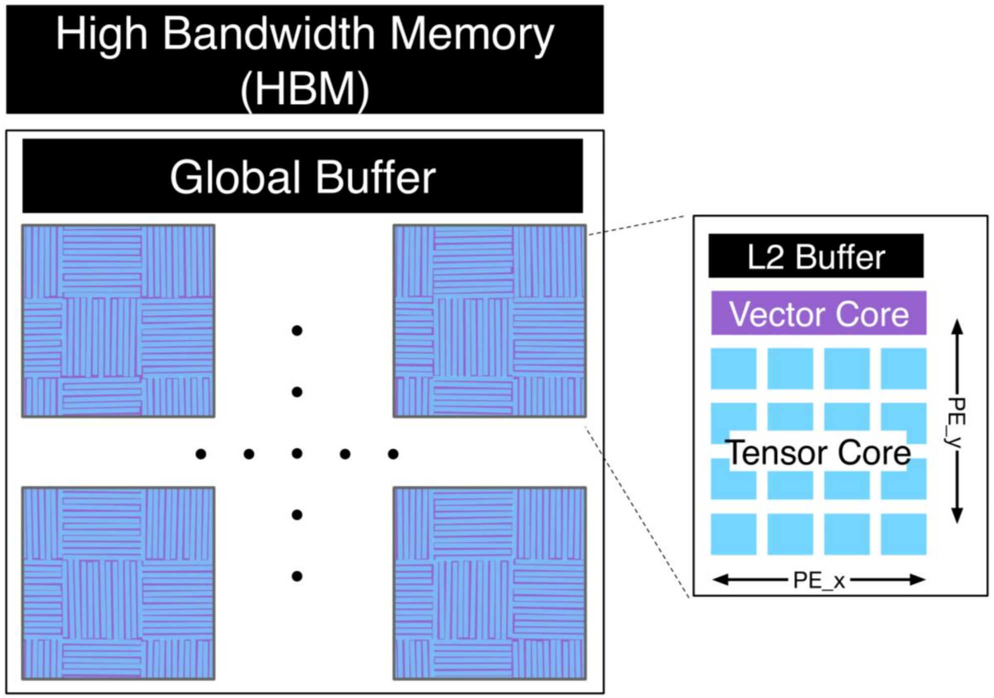
Overview of Phaze: Architecture Generator



Compute parameters: {num_{tc}, num_{vc}, PE_x, PE_y, Pe_{vc}}

Memory Parameters: {GLB, GLB_{bw}, $L2_{tc}$, $L2_{vc}$}, {HBM}

Overview of Phaze: Architecture Generator

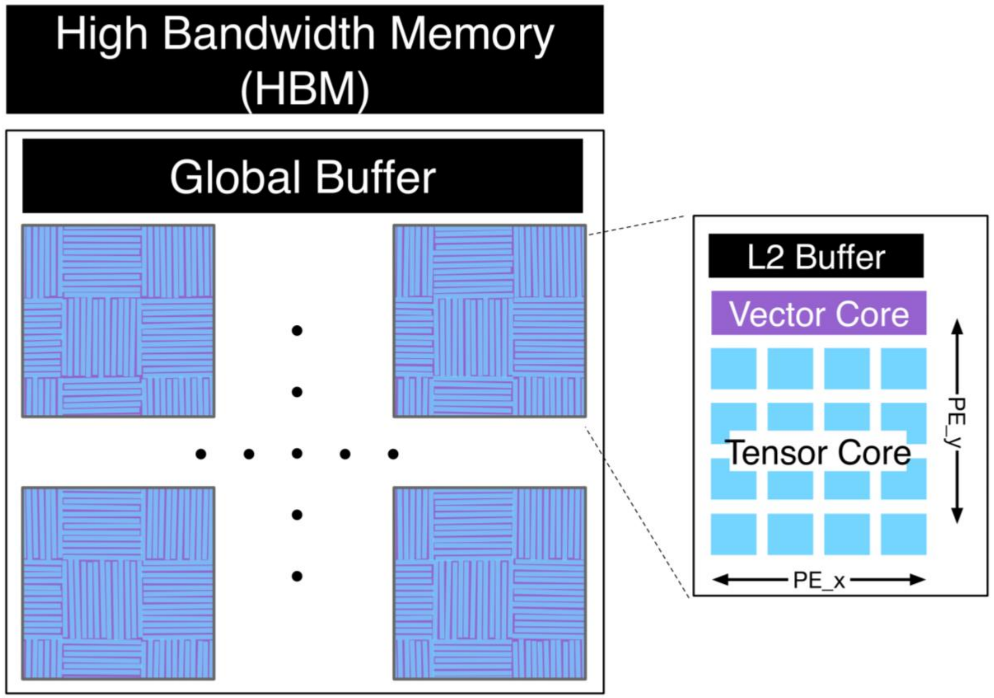


Area Estimation

Compute parameters: {num_{tc}, num_{vc}, PE_x, PE_y, Pe_{vc}}

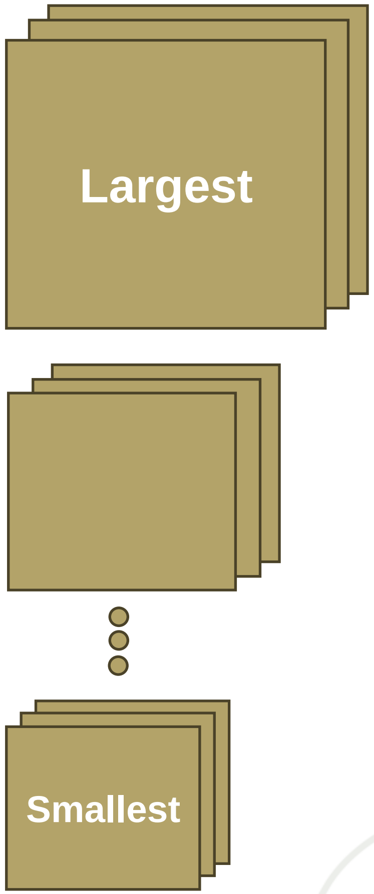
Memory Parameters: {GLB, GLB_{bw}, L2_{tc}, L2_{vc}}, {HBM}

Overview of Phaze: Architecture Generator



Area Estimation

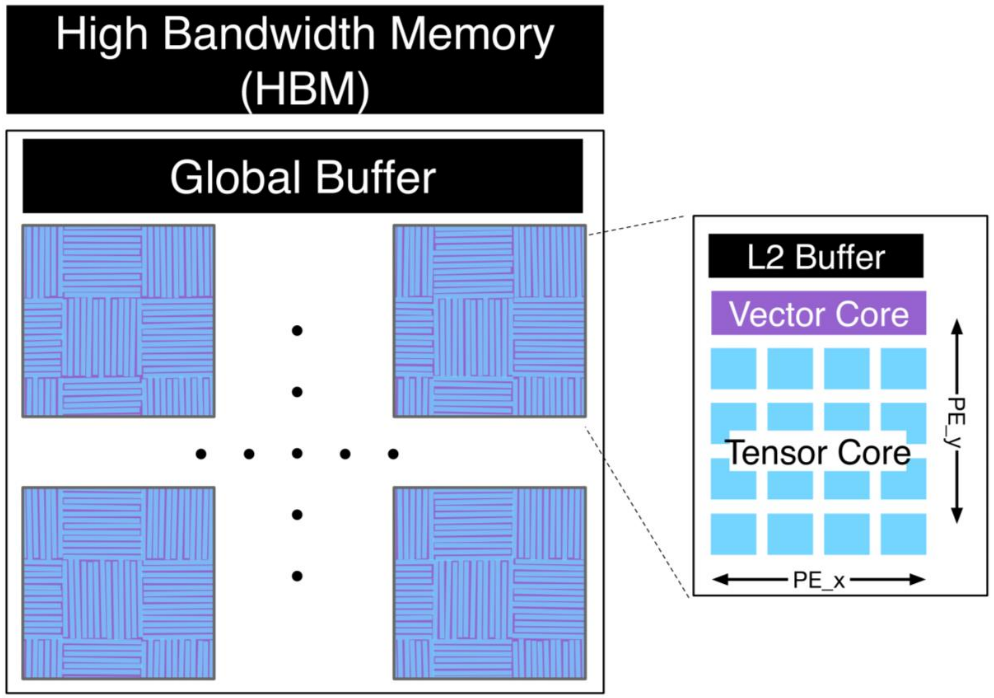
Rank by area



Compute parameters: {num_{tc}, num_{vc}, PE_x, PE_y, Pe_{vc}}

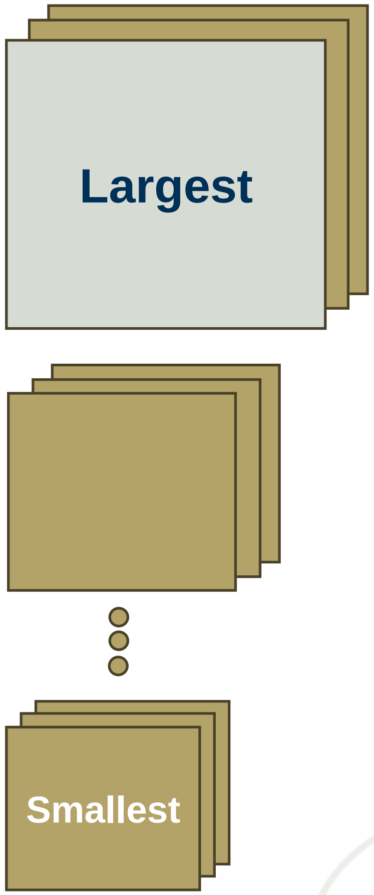
Memory Parameters: {GLB, GLB_{bw}, $L2_{tc}$, $L2_{vc}$}, {HBM}

Overview of Phaze: Architecture Generator



Area Estimation

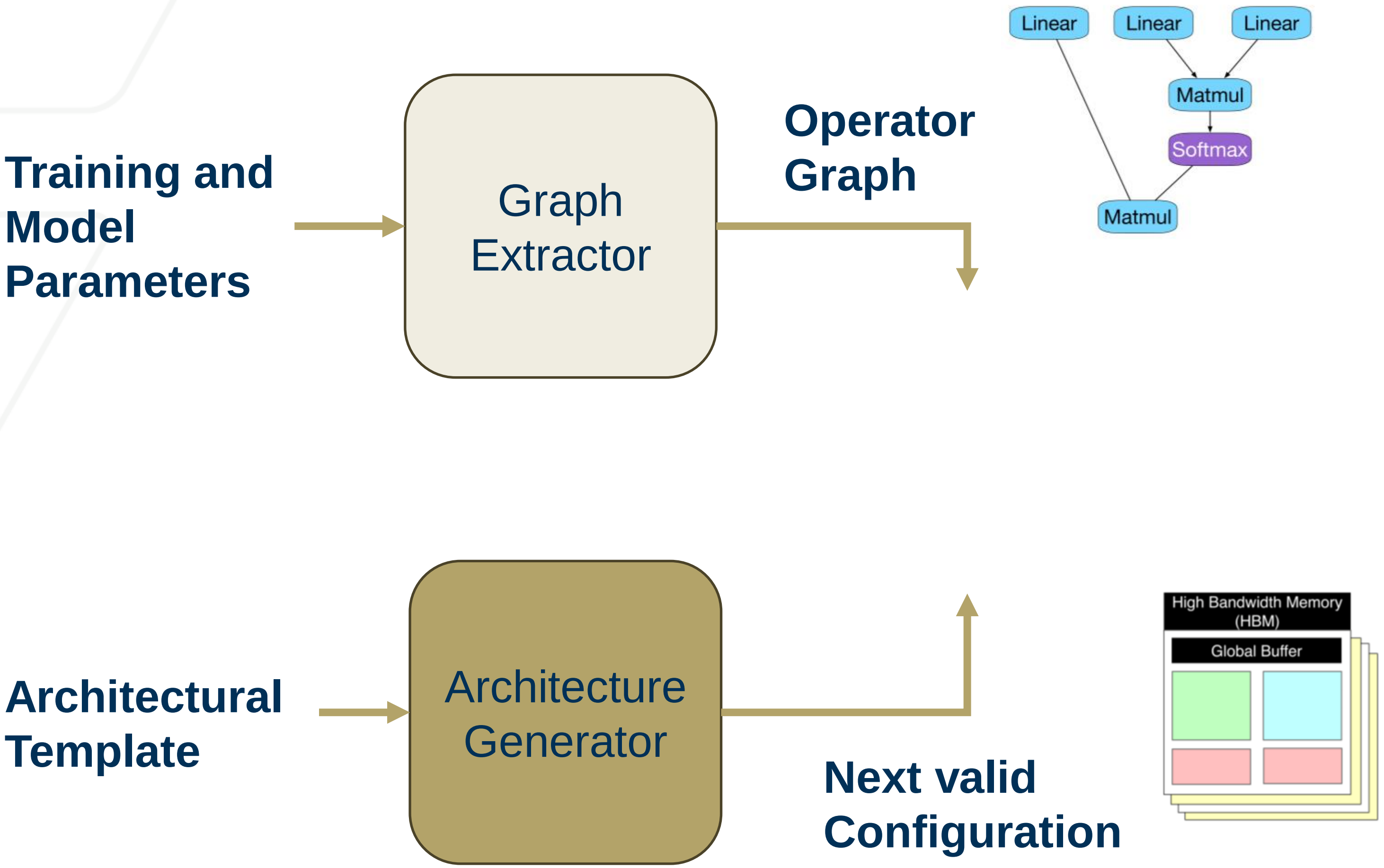
Rank by area



Compute parameters: {num_{tc}, num_{vc}, PE_x, PE_y, Pe_{vc}}

Memory Parameters: {GLB, GLB_{bw}, $L2_{tc}$, $L2_{vc}$}, {HBM}

Overview of Phaze



Training and Model Parameters

Graph Extractor

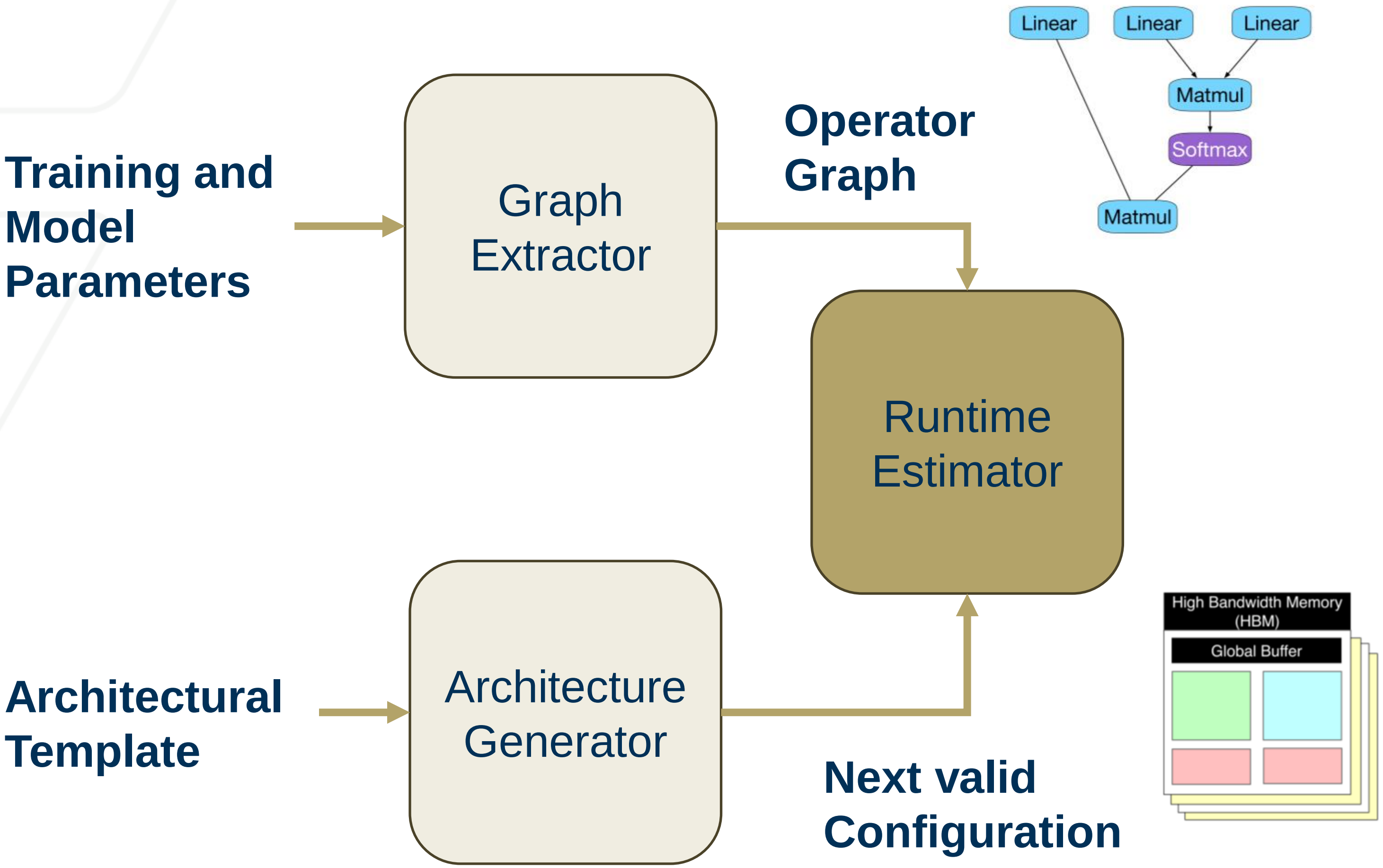
Operator Graph

Architectural Template

Architecture Generator

Next valid Configuration

Overview of Phaze



Training and Model Parameters

Graph Extractor

Operator Graph

Runtime Estimator

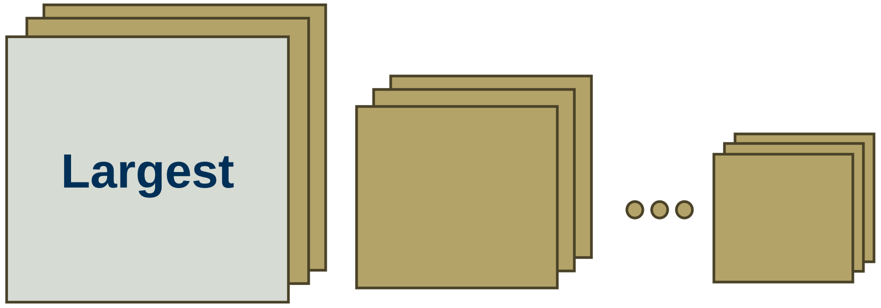
Architectural Template

Architecture Generator

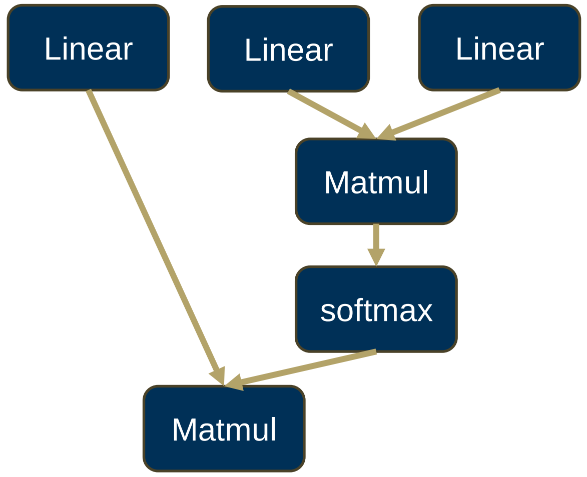
Next valid Configuration

Overview of Phaze: Estimator

Next Architecture Configuration

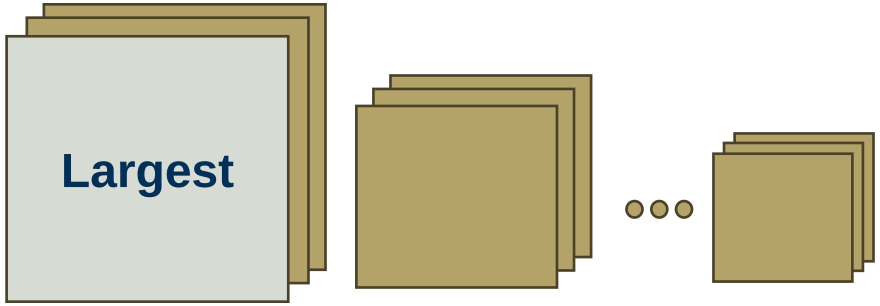


Operator Graph

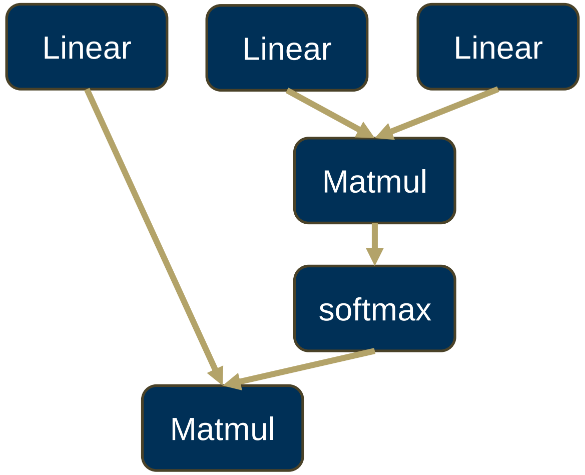


Overview of Phaze: Estimator

Next Architecture Configuration

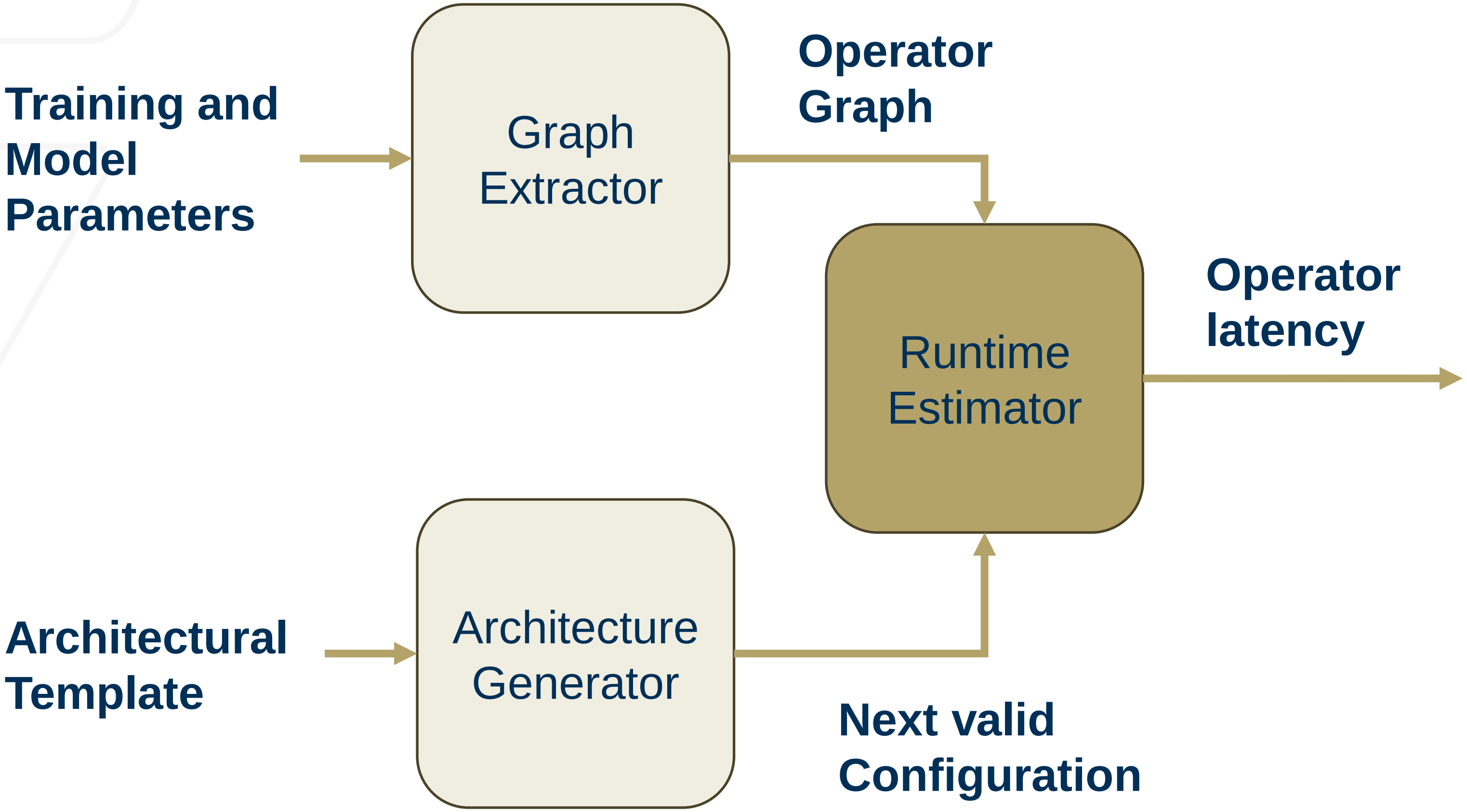


Operator Graph



Operator Latency Estimates

Overview of Phaze



Training and Model Parameters

Graph Extractor

Operator Graph

Runtime Estimator

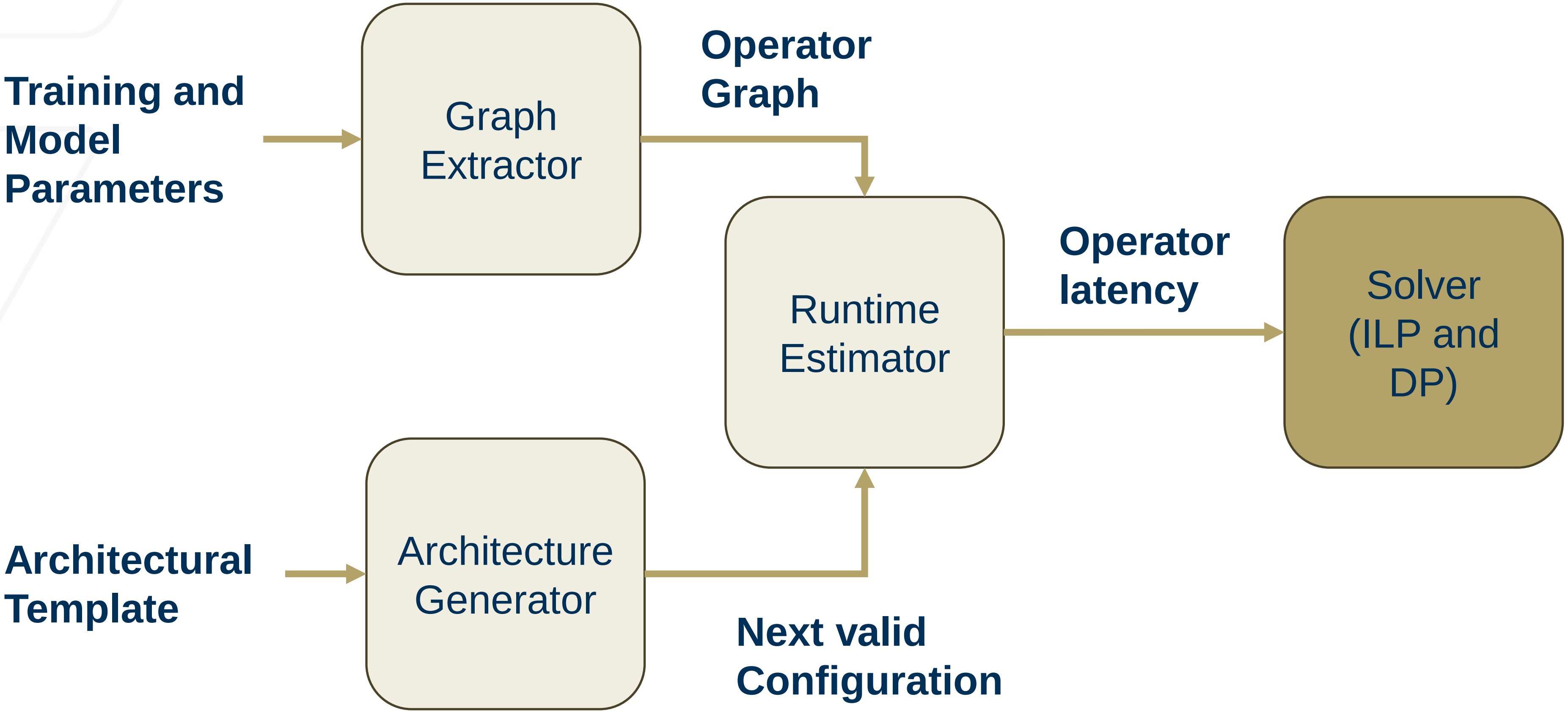
Operator latency

Architectural Template

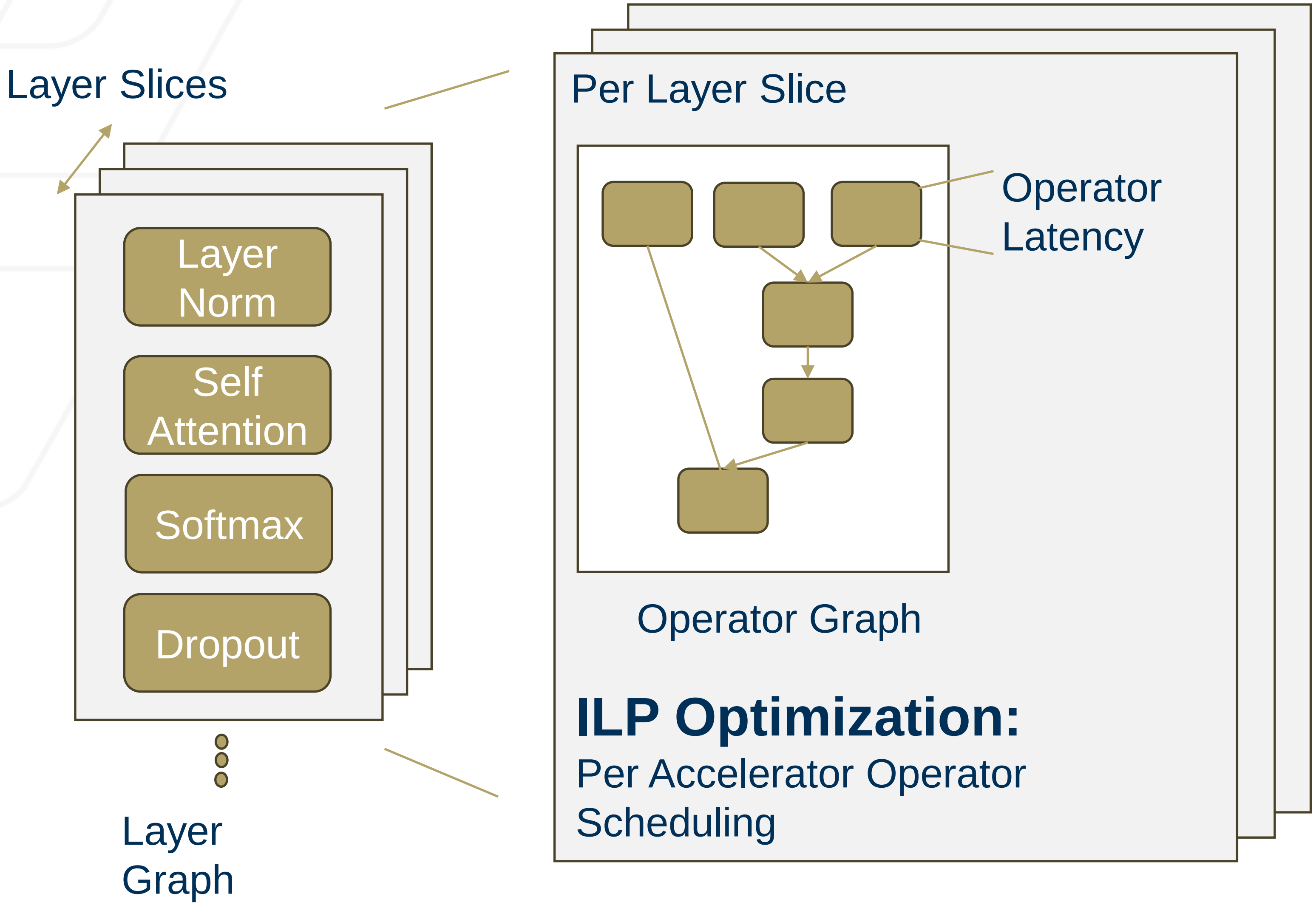
Architecture Generator

Next valid Configuration

Overview of Phaze



Overview of Phaze: Integer Linear Program

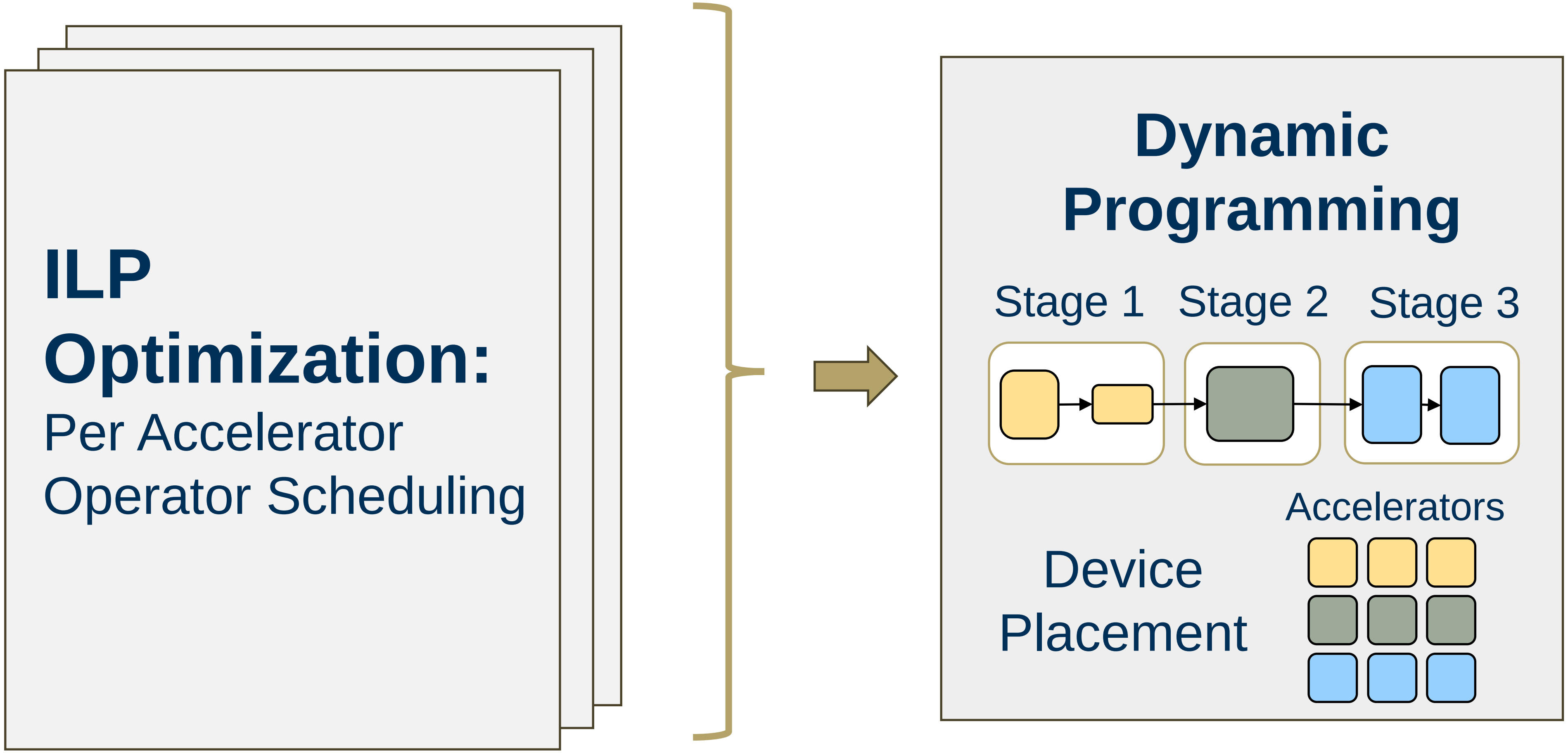


ILP Optimization:

Per Accelerator Operator Scheduling

- Constraints ensure:
  - Order and dependencies in graph
  - Resource constraints are met
- Reduced Complexity without time-indexed variables

Overview of Phaze: Dynamic Progamming



ILP Optimization:

Per Accelerator Operator Scheduling

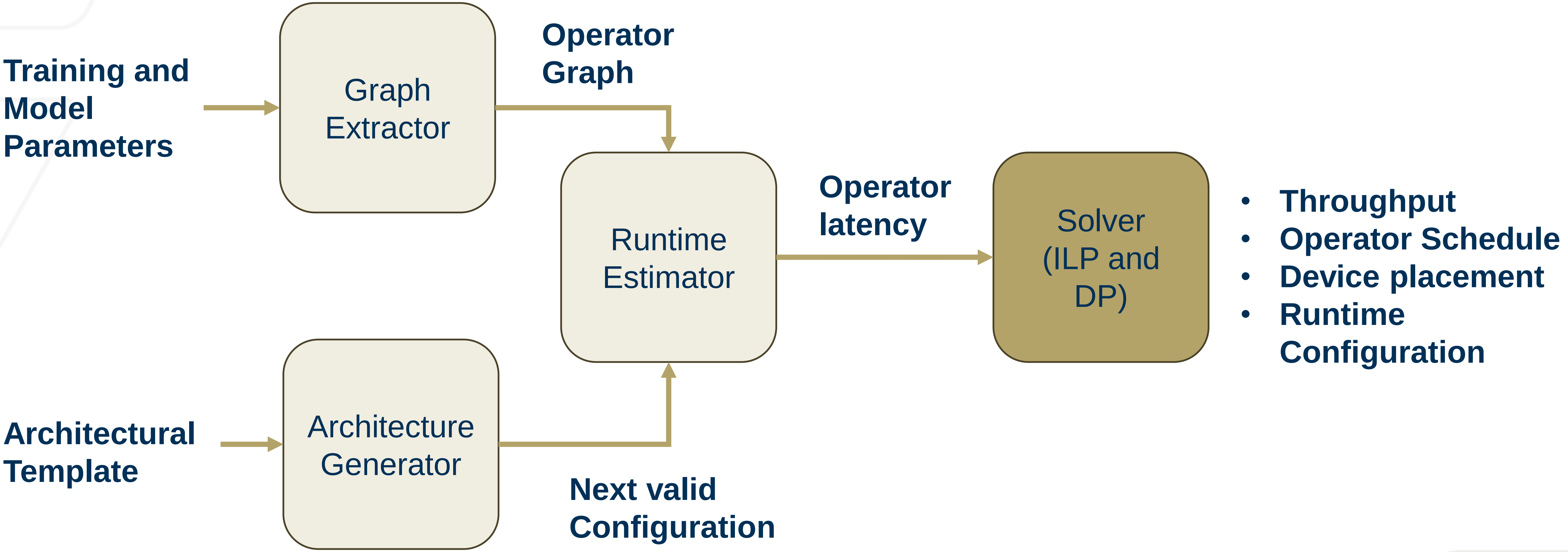
Dynamic Programming

Stage 1 Stage 2 Stage 3

Accelerators

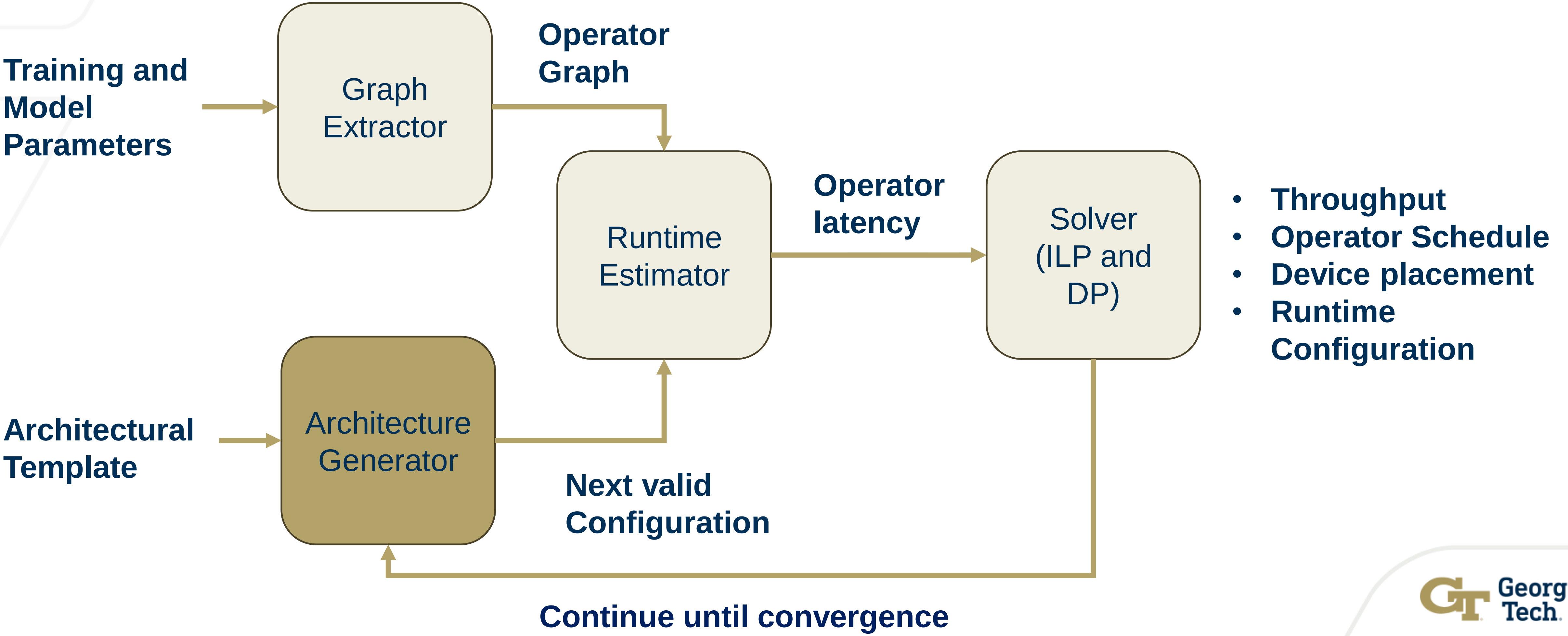
Device Placement

Overview of Phaze



- Throughput
- Operator Schedule
- Device placement
- Runtime Configuration

Overview of Phaze



Training and Model Parameters

Graph Extractor

Operator Graph

Runtime Estimator

Operator latency

Solver (ILP and DP)

- Throughput
- Operator Schedule
- Device placement
- Runtime Configuration

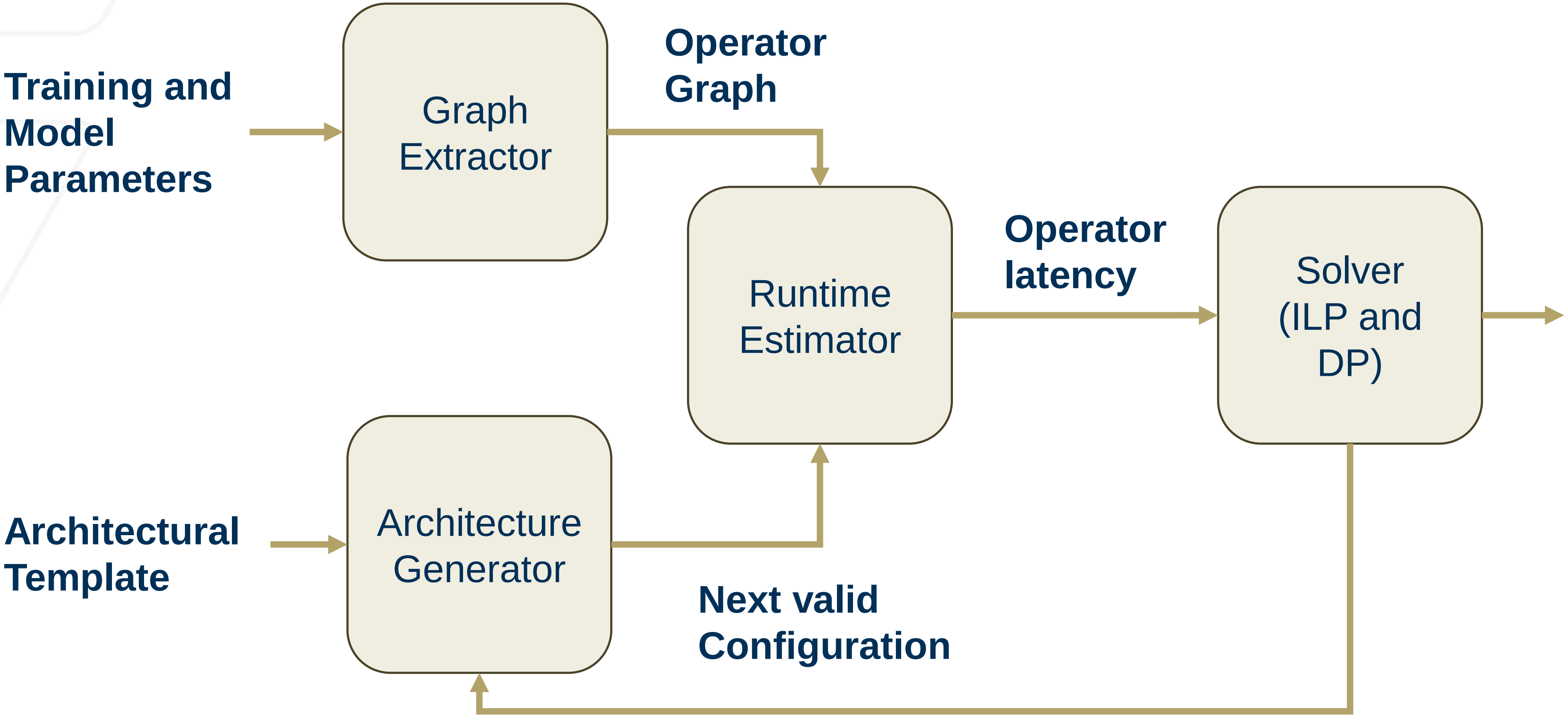
Architectural Template

Architecture Generator

Next valid Configuration

Continue until convergence

Overview of Phaze



At Convergence:

- Operator Schedule
- Device placement
- Runtime Configuration
- Architecture Configuration

Evaluations

Comparison baselines

Architecture Baselines:

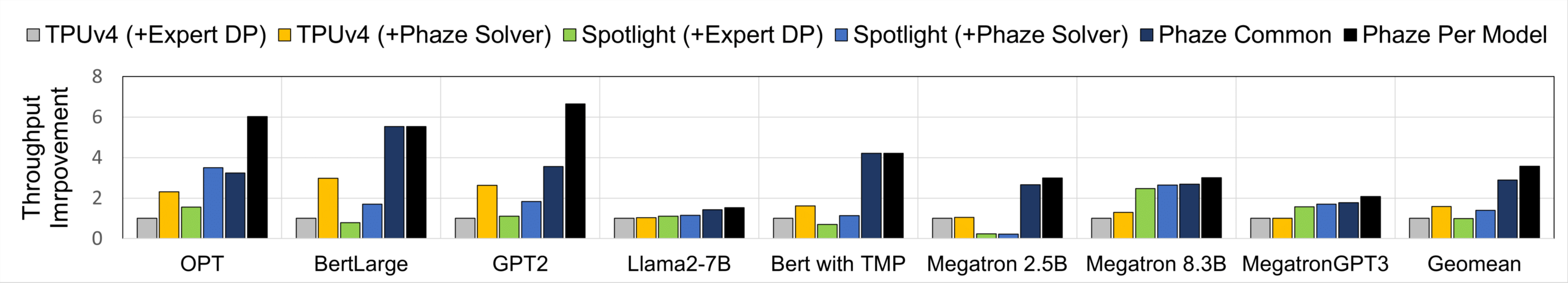
- TPUv4 architecture
- Spotlight- searched architectures

Each architecture is executed with:

- Fixed Expert device placement strategy
- Phaze solver device placement strategy

Throughput Results

Phaze **Model specific** and **Common** configurations on average provide **3.6x and 2.9x** higher throughput than TPUv4 architecture



Phaze Architecture Characteristics

- 91% area utilization
- Large Tensor Cores
- High number of Vector Cores
- < 64 GB HBM

Conclusion

Phaze is an algorithmic solution for distributed training:

- Co-optimization between architecture search and device placement
- Novel ILP programs that reduces convergence time
- Makes the multi-dimensional search space tractable
- Achieves higher throughput compared to state-of-the-art solutions

Future Work

- Adding New Evaluation Metrics to Phaze
    - Carbon
    - Power
    - Cost
- Adding Support to model more realistic networks
    - Current Assumes a flat network
    - More sophisticated collective communication modelling