NEURAL INFORMATION PROCESSING SYSTEMS

# ASPEN:

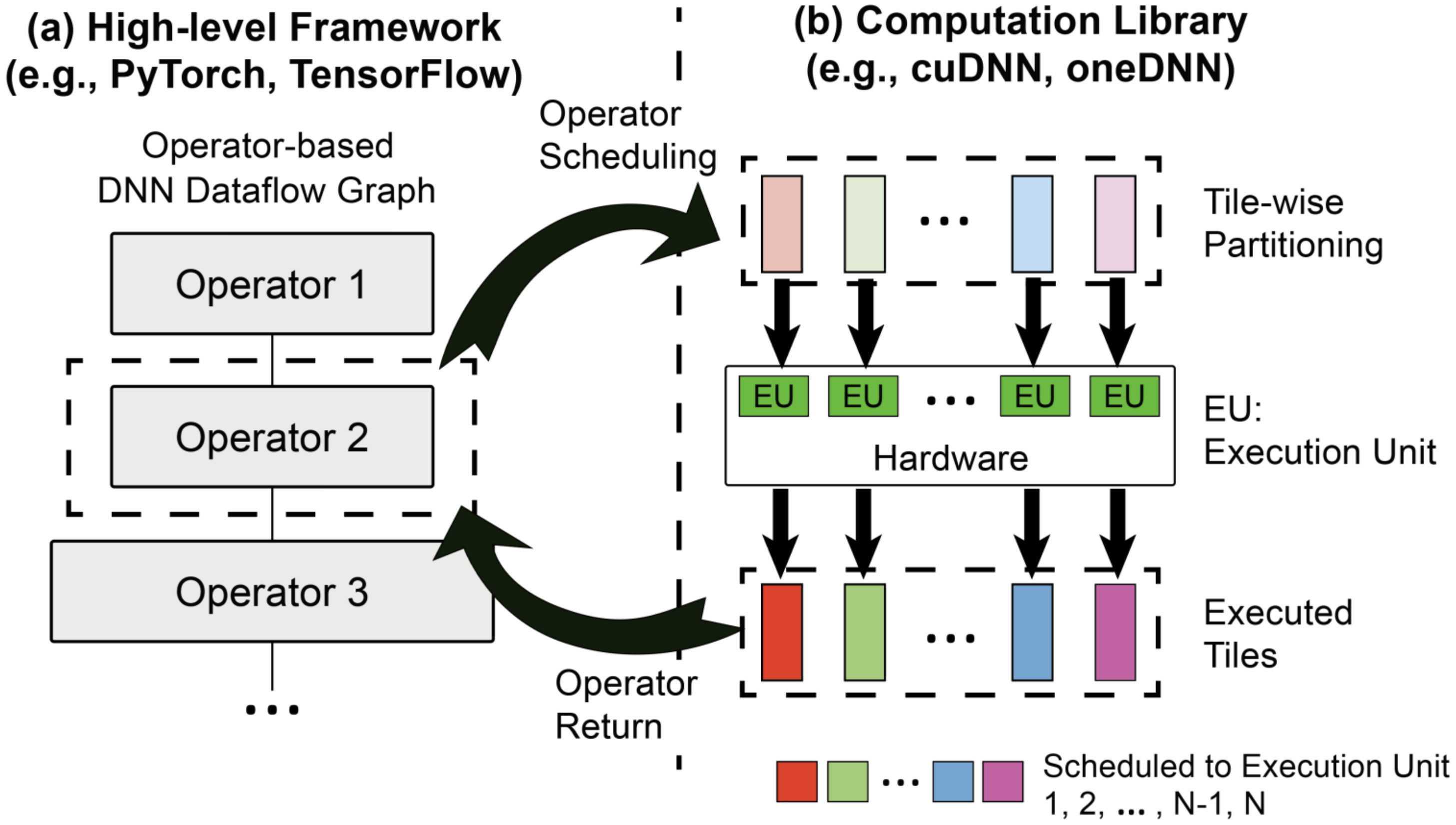
## Breaking Operator Barriers for Efficient Parallel Execution of Deep Neural Networks

Jongseok Park[1]
with Kyungmin Bin[1], Gibum Park[1], Sangtae Ha[2], and Kyunghan Lee[1]
[1]Seoul National University, [2]University of Colorado Boulder
NeurIPS 2023, December 10th – 15th, New Orleans, Louisiana

NXC LAB

# Current Computation of DNNs

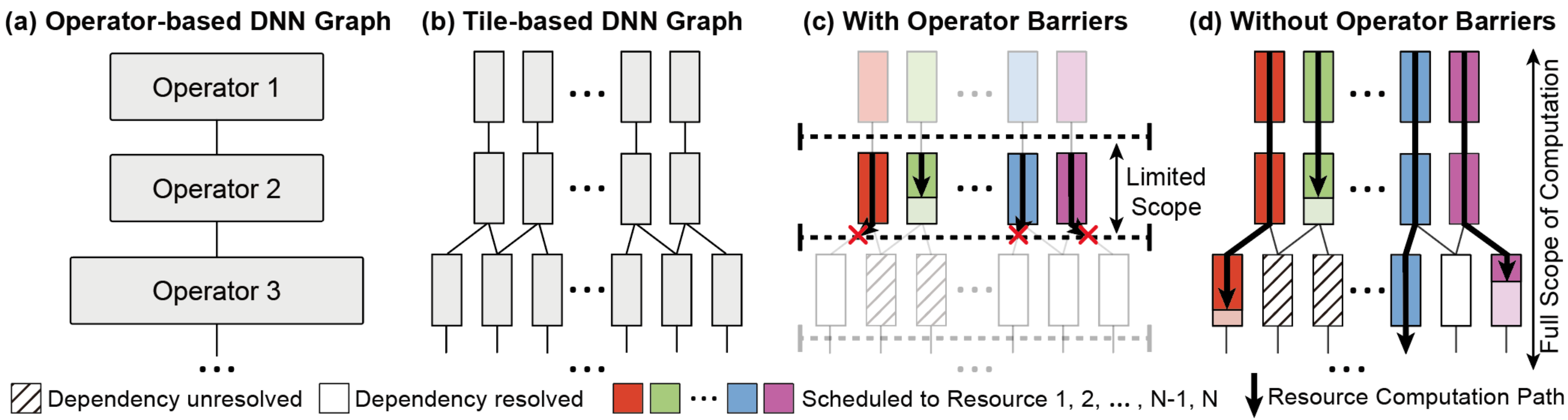
- Current Solution: Operator-based Two-level Computation
  1. **High-level frameworks** (e.g., PyTorch) schedule operators to **Computation Libraries.**
  2. **Computation Libraries** (e.g., cuDNN) execute operators in parallel, using **Tiles**.




❖ **Tiles**
Unit of parallel computation. Operators are partitioned into tiles, and scheduled to parallel execution units.

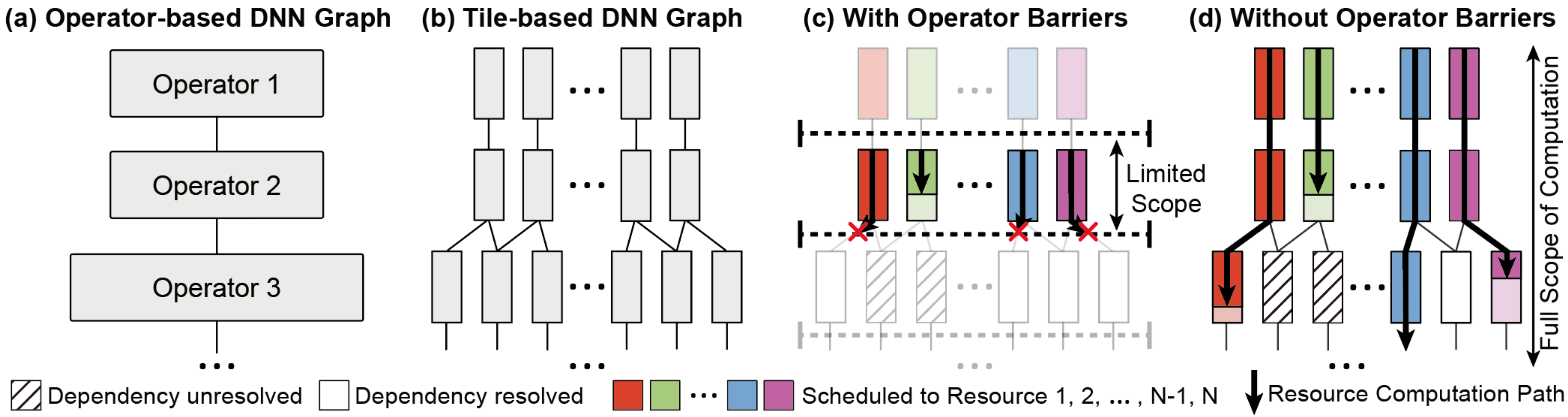
# Problem of Operator-based Dataflow Graphs

- Operators require Synchronization barriers!
  - Synchronization barriers are used to ensure all tiles are executed.
  - Barriers guarantee the satisfaction of dependency before executing the next operator.
  - However, barriers also isolate the scope of computation within an operator.



# Breaking the Operator Barriers

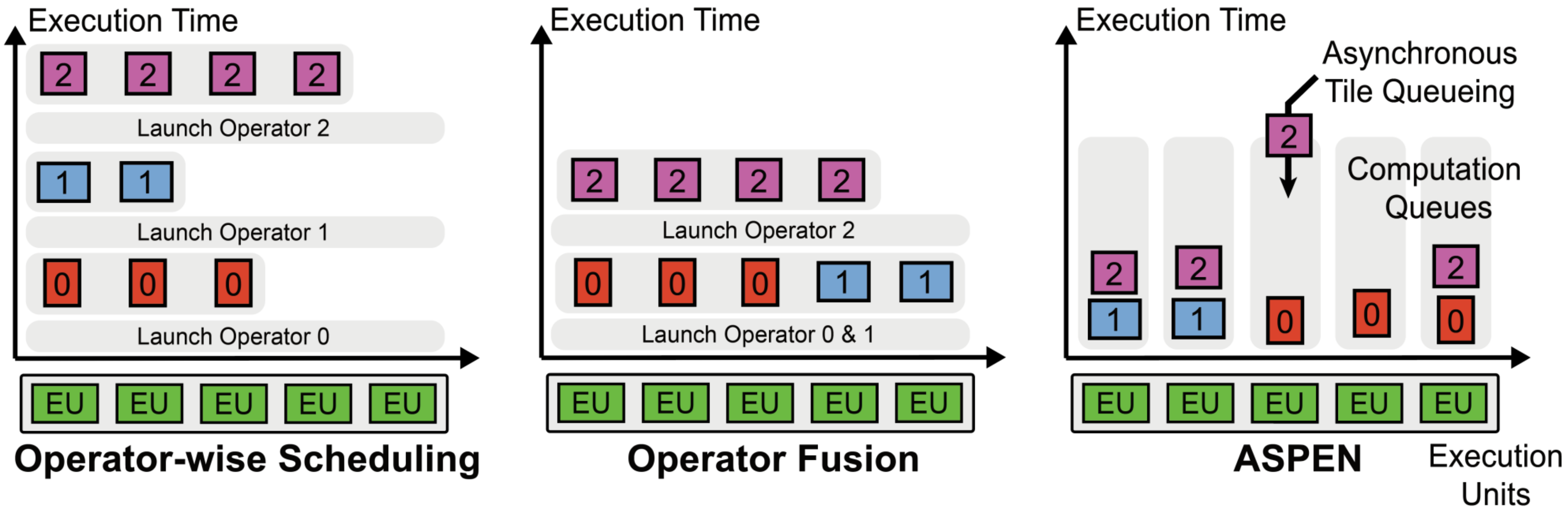
- Breaking the operator barriers reveals rich dataflow relationships
  - Rich dataflow relationships between tiles are hidden behind operator boundaries.
  - Breaking (removing) operator barriers reveals these rich relationships.
  - Rich dataflow relationships expose new opportunities!
    - New parallel computations, data reuses, asynchronous executions…



# Main idea of ASPEN

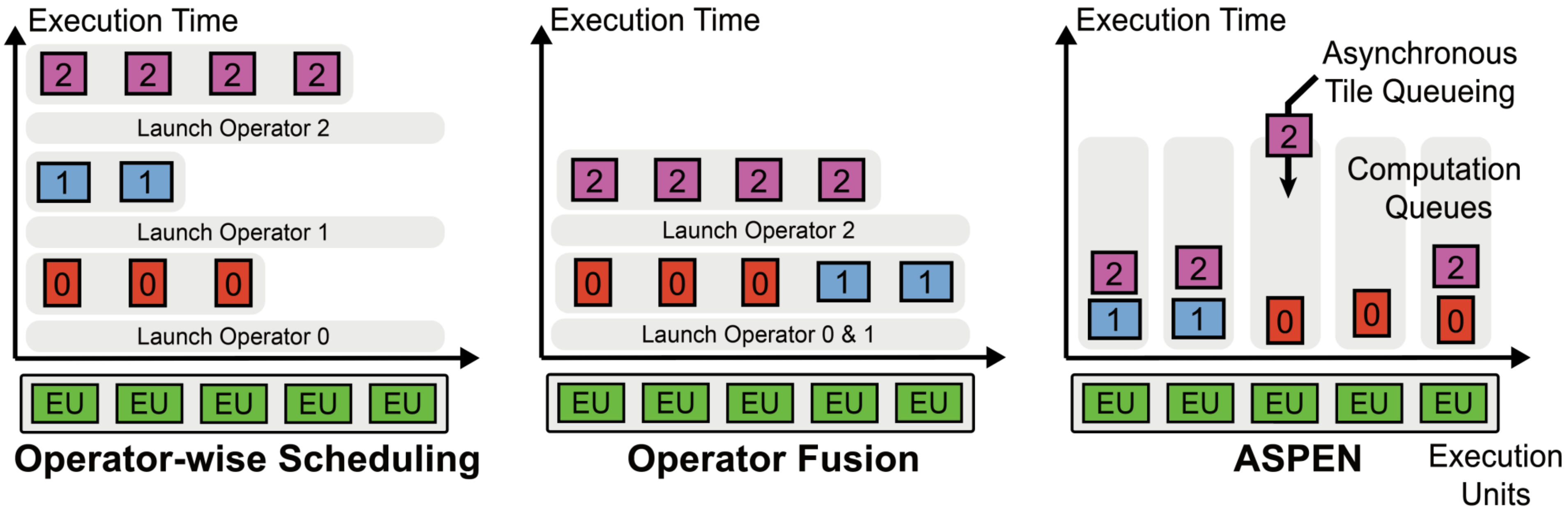
- A system that can dynamically utilize novel computational opportunities!
  - Break synchronization barriers to expose fine-grained computational opportunities
  - Each resource dynamically tracks and identifies computation opportunities during runtime
  - Asynchronously schedule and execute the opportunities for maximum utilization

> ➢ "Using fine-grained dynamic execution of DNNs, ASPEN achieves **Opportunistic Parallelism**"



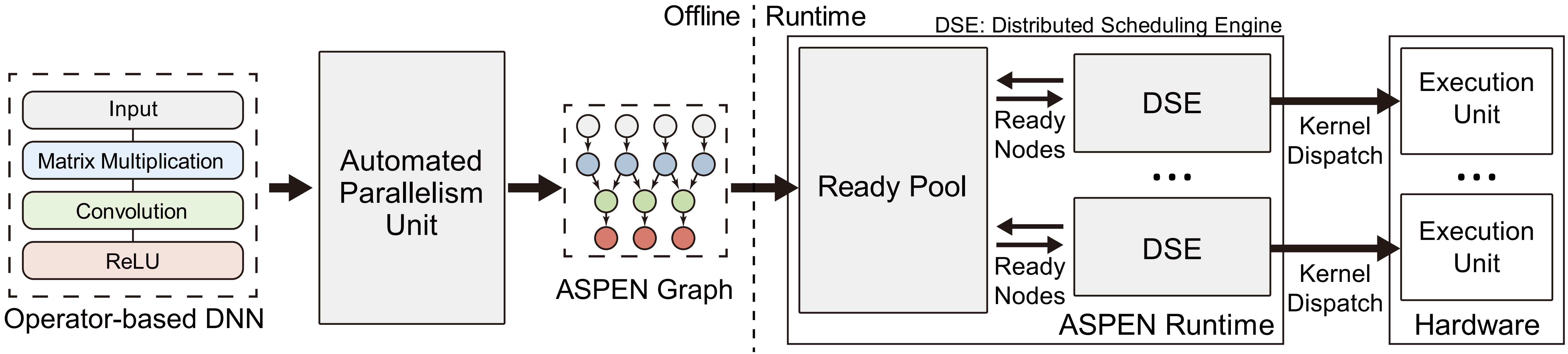
# Benefits of ASPEN

- ASPEN's novel execution brings following benefits…
  1. Graph-wide scope of parallel scheduling enables maximal parallelism
  2. Dynamic, distributed runtime improves utilization, scalability, and load-balancing
  3. Asynchronous execution interleaves of computation, data movement, and scheduling
  4. Novel depthwise scheduling approach increases data reuse



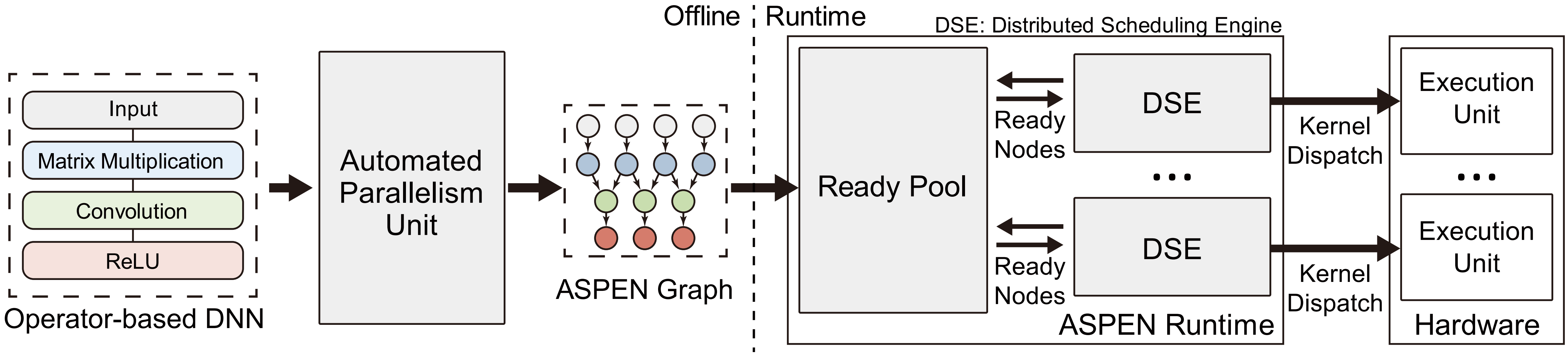
# ASPEN System Design

- Design challenges in ASPEN
  - How to break the barriers in a generalized way, applicable to any DNNs and operators?
  - How to create a runtime that can dynamically exploit the opportunities of fine-grained DNNs?
  - How can the asynchronous system share execution information, for correctness of the DNN?



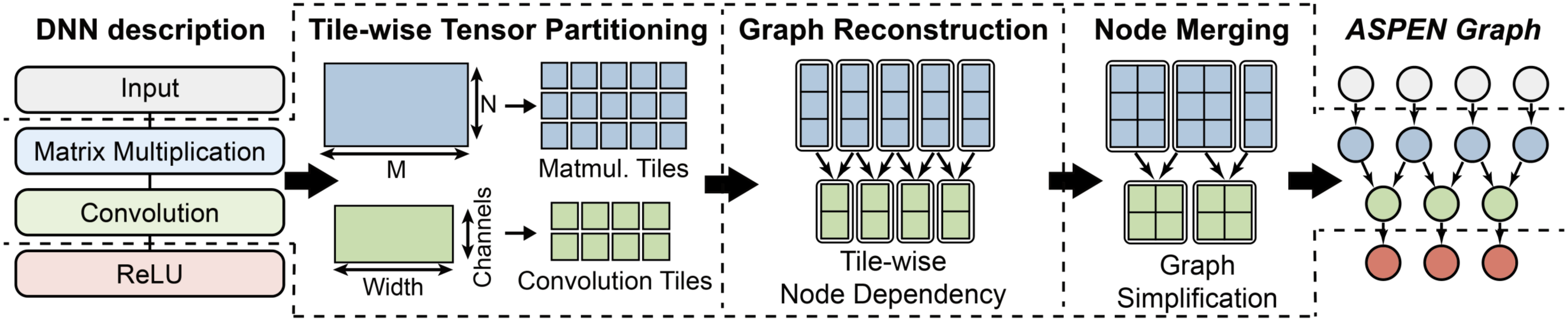
# ASPEN System Design

- System Component Overview
  - **Automated Parallelism Unit** (APU) – Parses DNNs into tile-based graphs.
  - **Distributed Scheduling Engine** (DSE) – Traverse & schedules graph nodes (tiles)
  - **Ready Pool** – Stores graph nodes that are ready for execution (DNN agnostic)



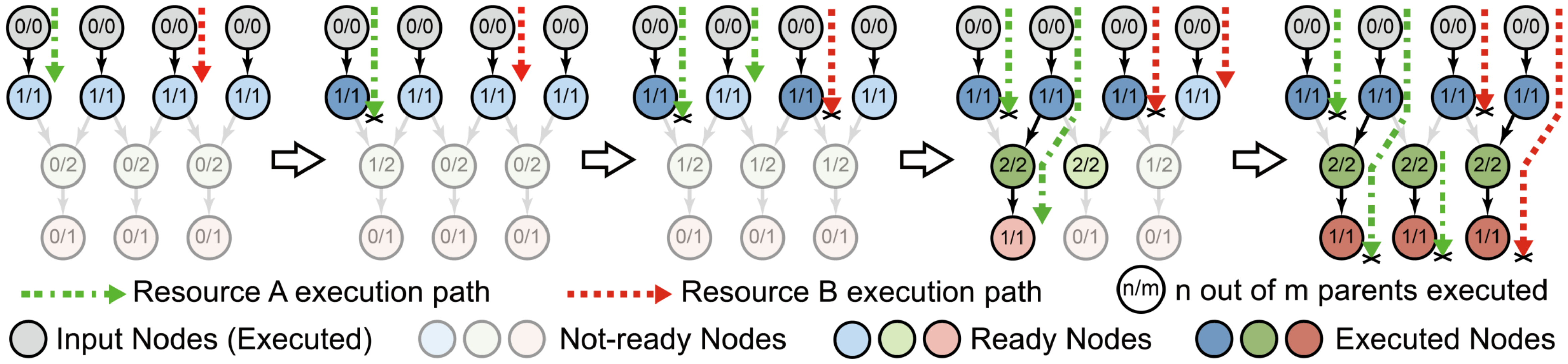
# ASPEN System Design

- Automated Parallelism Unit (APU)
  - **Parses** operator-based DNN description and partitions each operator output into tiles
  - **Graphs** the DNN into tile-based dataflow graph based on tile-wise dependency
  - **Merges** nodes into larger nodes (tiles) to simplify graph construction and increases per-tile kernel efficiency.



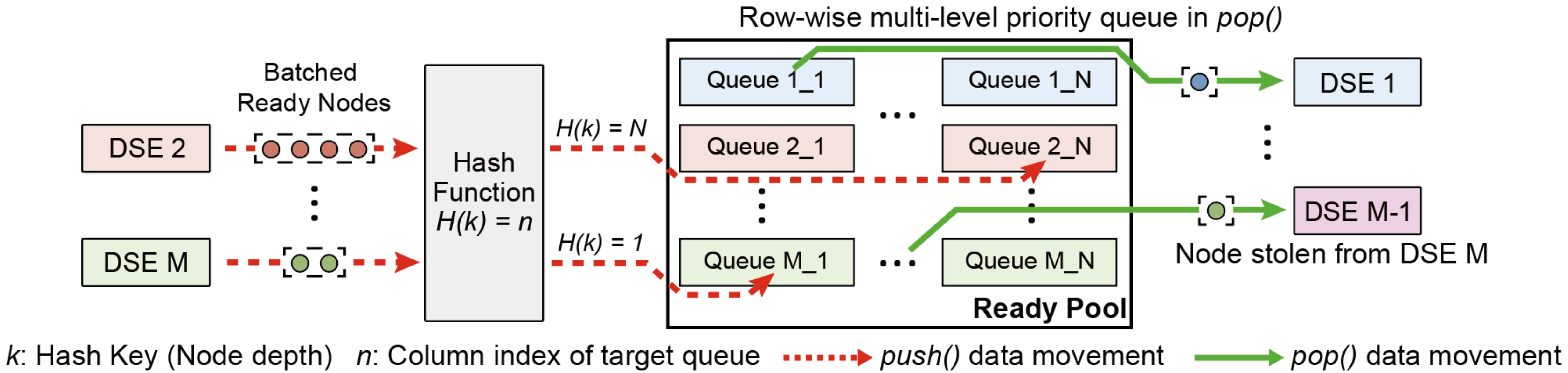
# ASPEN System Design

- Distributed Scheduling Engine (DSE)
  - **Exists per execution unit** – Higher scalability.
  - **Asynchronous graph traversal** – Maximal workload scheduling.
  - **Isolation of resources** – Less communication, higher utilization.
  - **Depth-first execution** – For increased data reuse.



# ASPEN System Design

- Ready Pool
  - Stores **dependency-satisfied** nodes (tiles), regardless of the origin DNN.
  - Uses a **matrix of concurrent queues** where each row is prioritized to each DSE.
  - DSE can **steal nodes** from other DSE's rows if its row is empty. (load-balancing)
  - **Scheduling policies** provided using hash function and priority queue accesses.

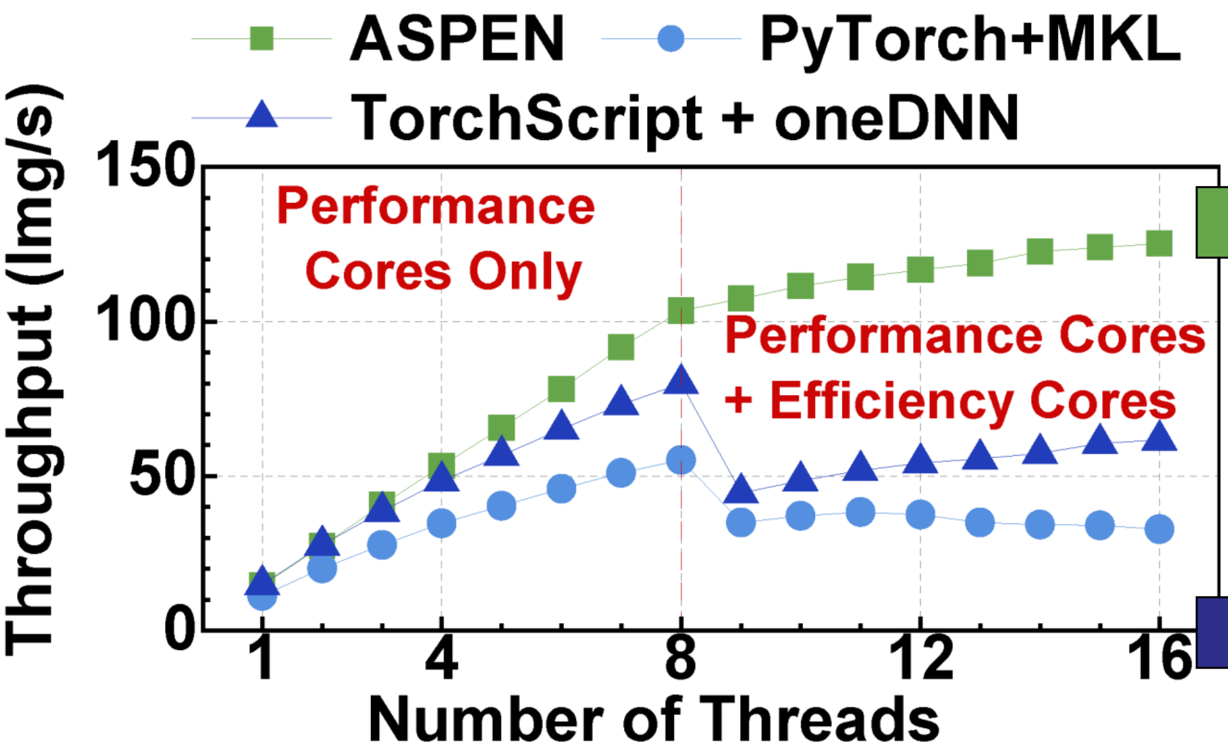


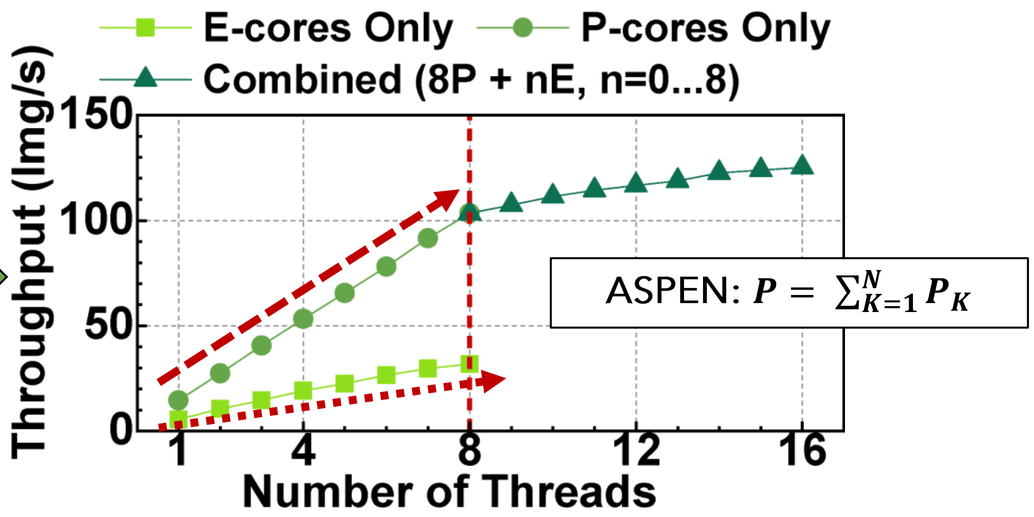


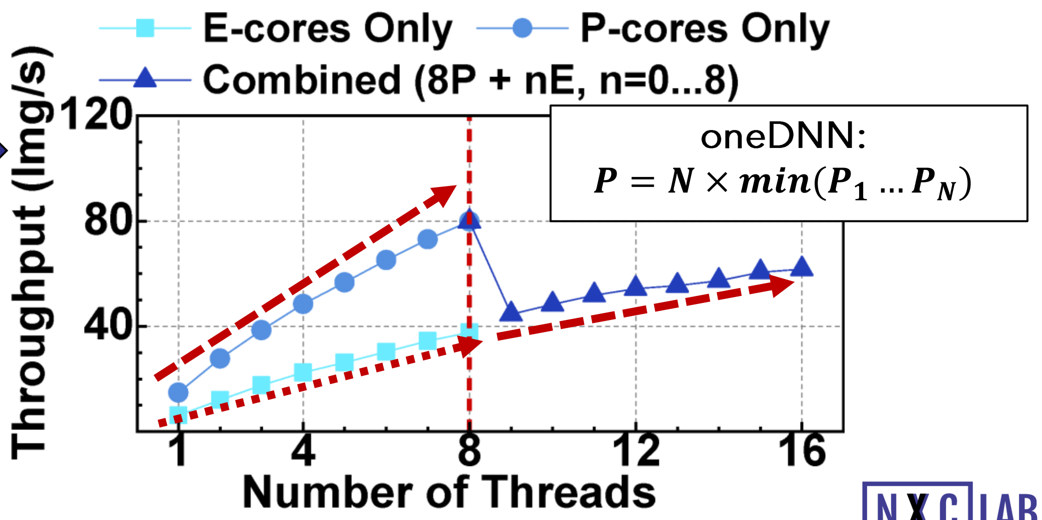
NXC LAB

# Evaluation (ResNet-50, Batch = 32)

- Dynamic Adaptation (Intel i9-12900K, 8 Performance + 8 Efficiency Cores)




ASPEN can dynamically utilize all available parallel resources to their full potential!

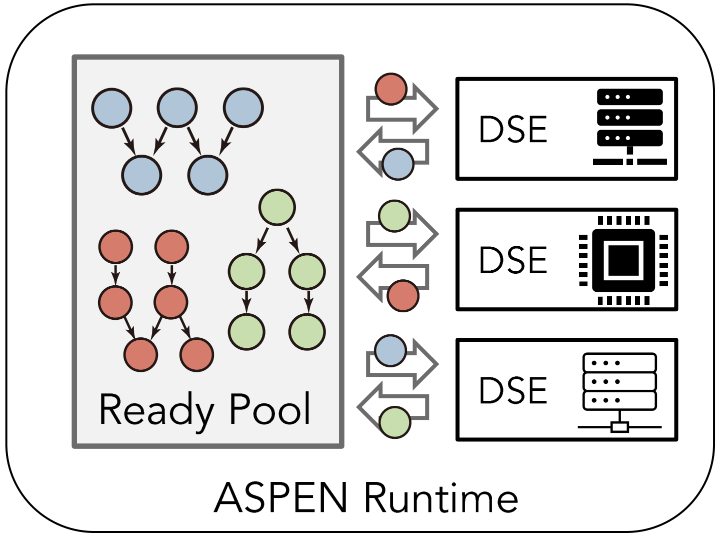
# Applications of ASPEN

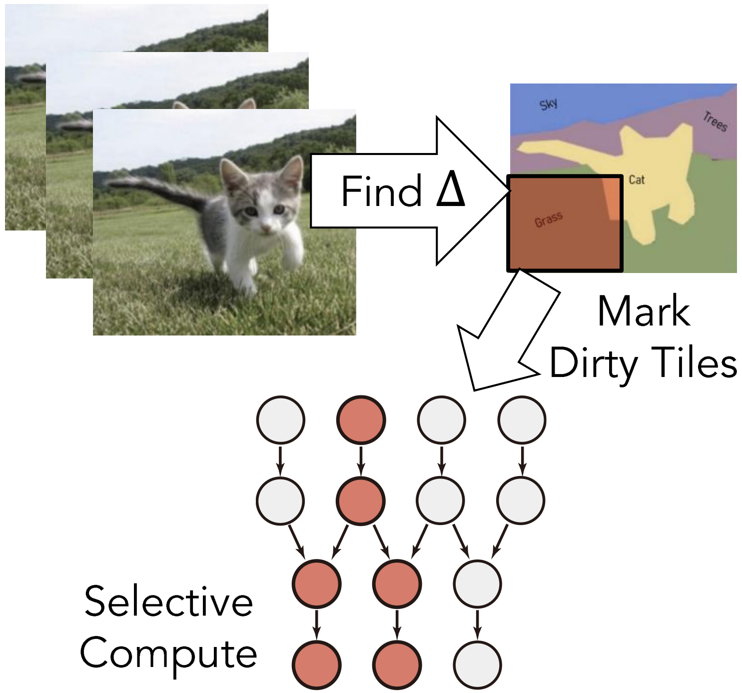
- The benefits of ASPEN is not only limited to performance!

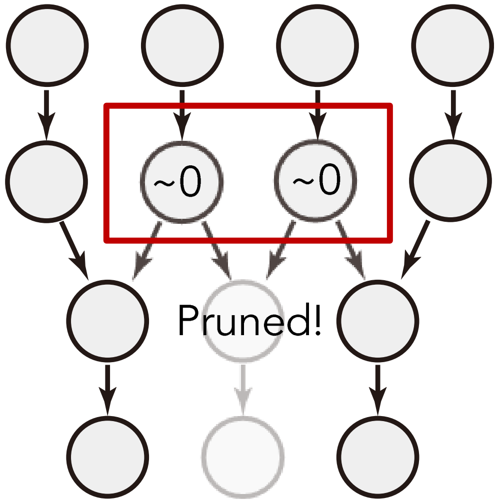
Out-of-the-box multi-tenant execution of different DNNs

Executing only the changed portion of DNN in a video stream

Dynamic pruning of DNNs



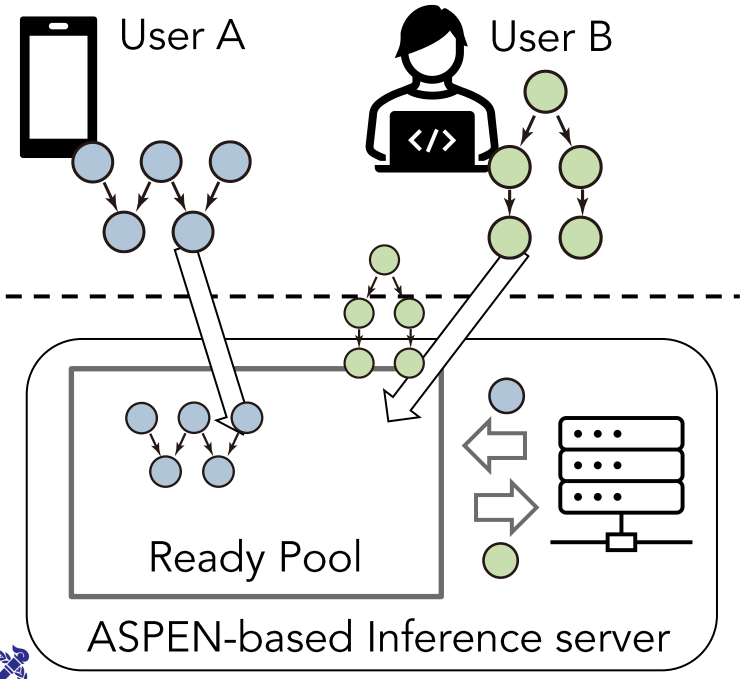
# Applications of ASPEN

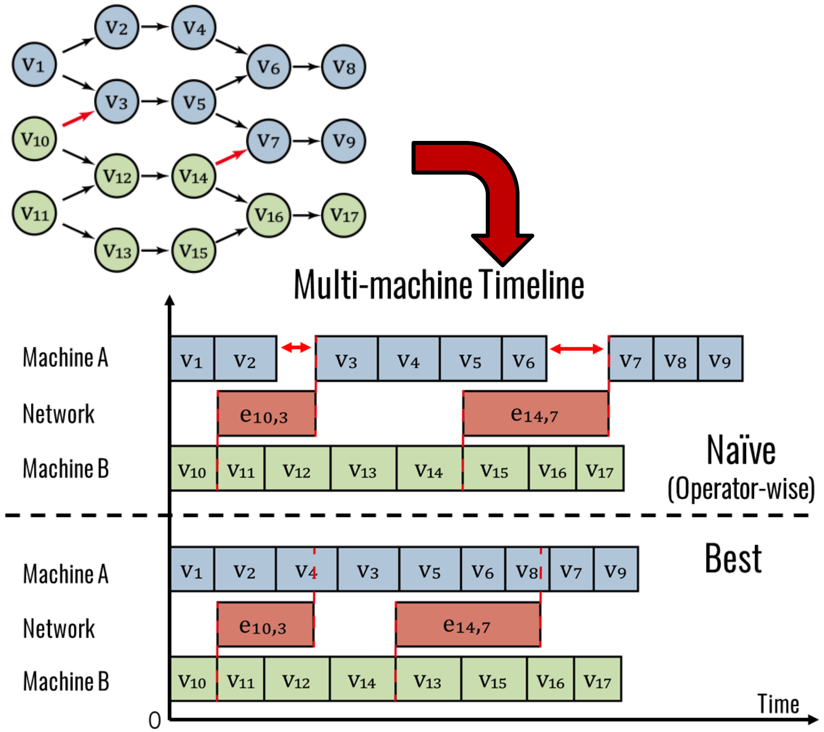
- Also, in networked computing scenarios such as in Inference Servers…

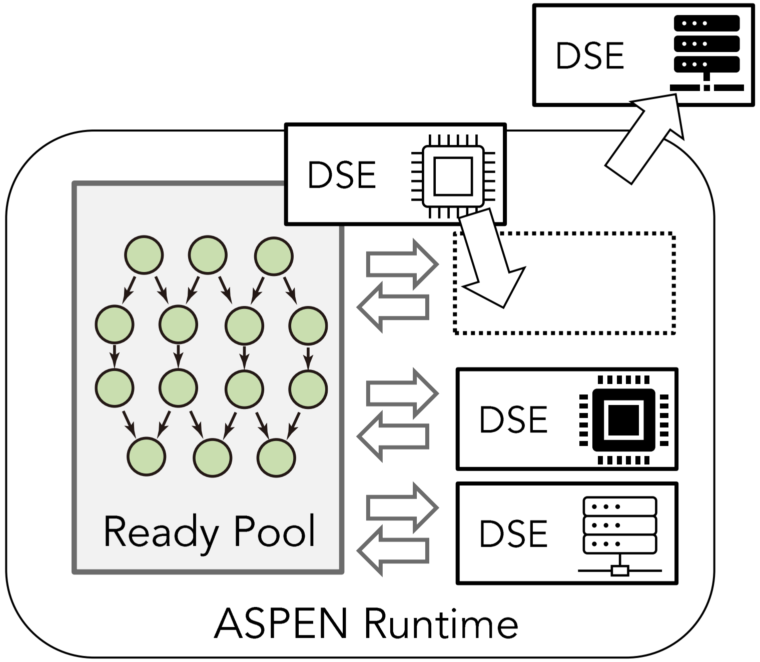
Interleaving of multiple DNN inference computation and communication

Fine-grained multi-machine scheduling of DNNs

Dynamic addition and removal of parallel resources while execution



# Thank you!

Presenter – Jongseok Park
cakeng@snu.ac.kr

Paper Link