LLMServingSim 2.0

A Unified Simulator for Heterogeneous and Disaggregated LLM Serving Infrastructure

Jaehong Cho†
Hyunmin Choi†
Guseul Heo
Jongse Park

KAIST

†Equal Contribution

KAIST

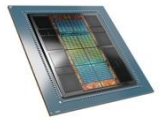
ISPASS

An Expanding LLM Serving Landscape

- Modern LLM serving spans a rapidly growing design space
- **Driven by heterogeneous hardware and diverse serving techniques**

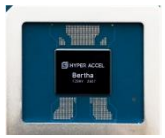
Hardware Heterogeneity

NVIDIA

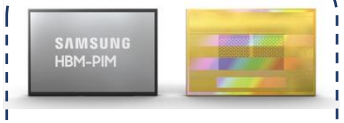


AMD

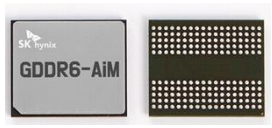
FURIOSA



HYPER ACCEL



SAMSUNG



SK hynix

GPU NPU PIM

Serving Technique Diversity

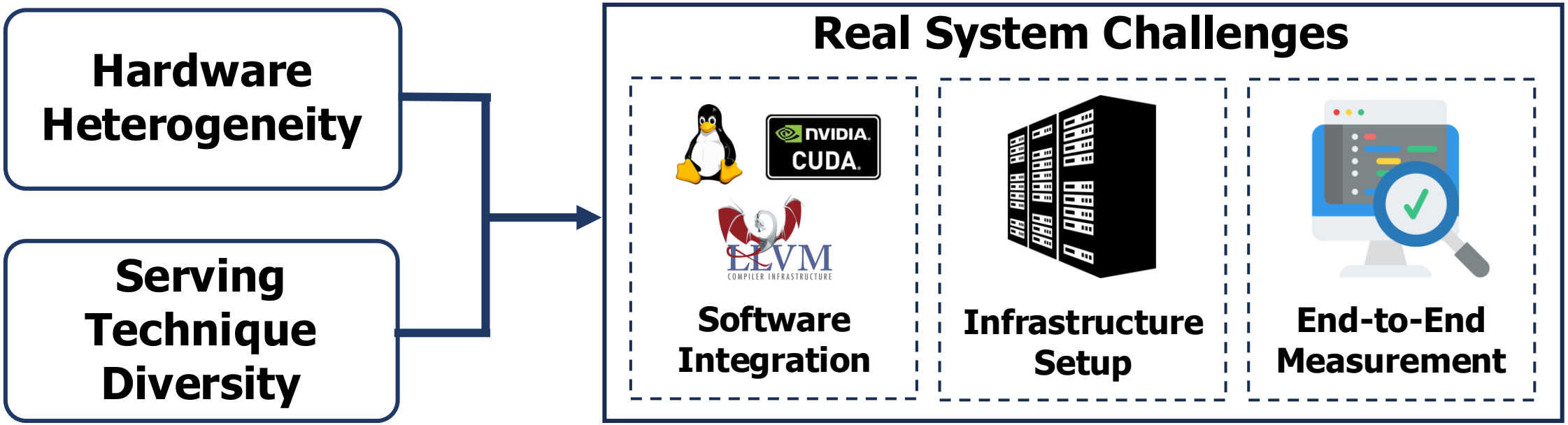
vLLM
- Expert Offloading
- P/D Disaggregation
- Prefix Caching

llm-d
- Multi-Instance
- Request Routing

LMCache
- KV Cache Offloading
- KV Cache Sharing

Rethinking Exploration for LLM Serving

- Real-system validation is **costly, engineering-heavy, and often impractical**
  - Even one serving design may require substantial software, infrastructure, and testing effort

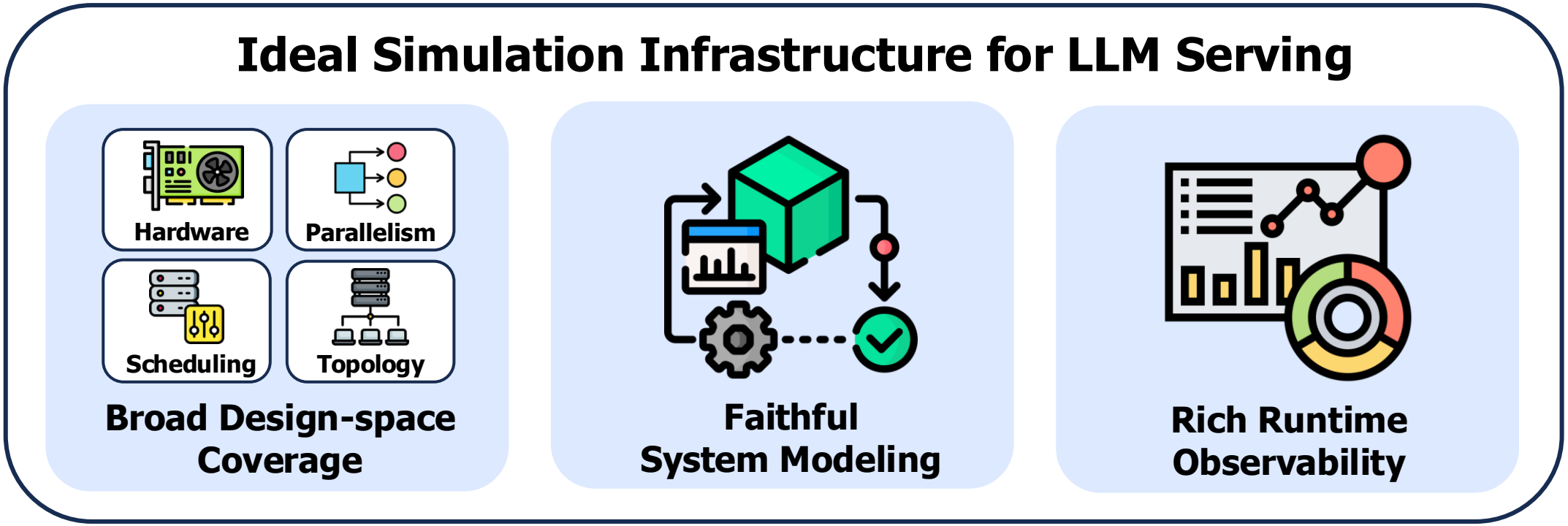


Rethinking Exploration for LLM Serving

- **Simulation enables practical exploration of diverse serving behavior**
  - It can estimate system behavior and performance with high fidelity before real deployment

Requirements for LLM Serving Simulator

- **A useful simulator should cover the full design space of LLM serving**
  - This requires broad support for disaggregation, parallelism, and serving-specific modeling

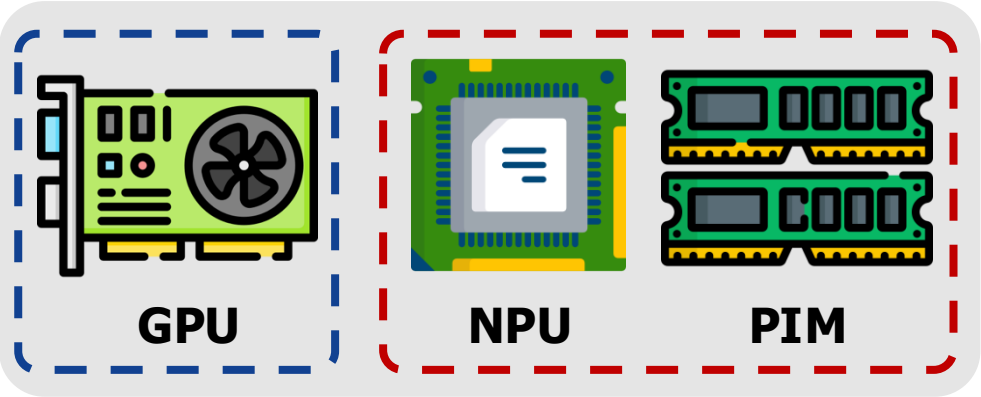


What Current LLM Simulators Still Miss

Existing LLM Serving Simulators

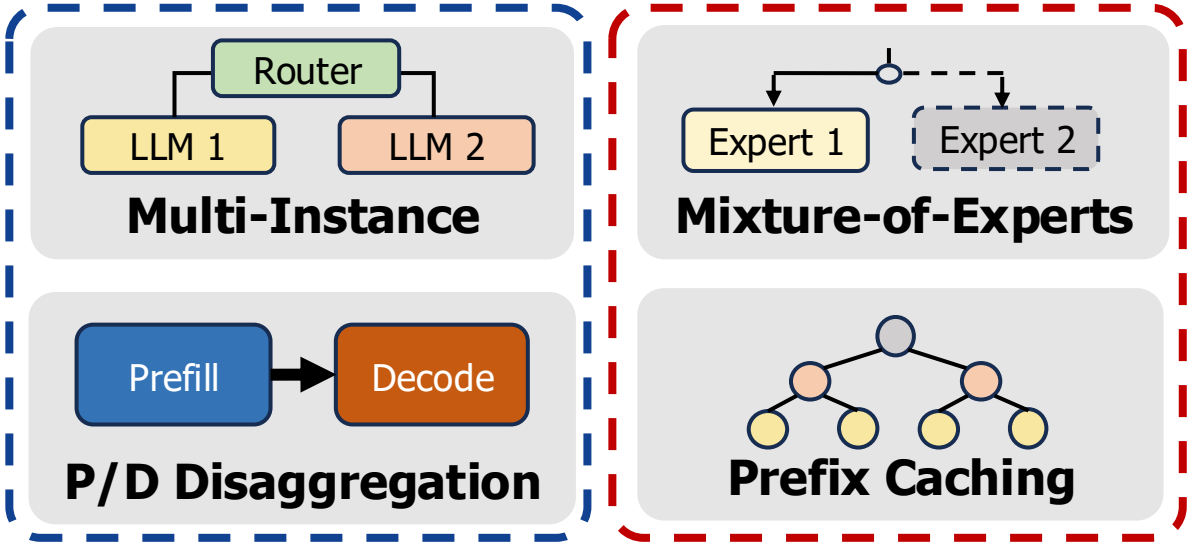
LLMServingSim [IISWC'24], Vidur [MLSYS'24], LLMCompass [ISCA'24], ADOR [ISPASS'25], etc.

Hardware Heterogeneity



- Largely **GPU-centric**
- Little support for emerging or heterogeneous hardware

Serving Technique Diversity



- Coverage remains limited to **a narrow subset**

What Current LLM Simulators Still Miss

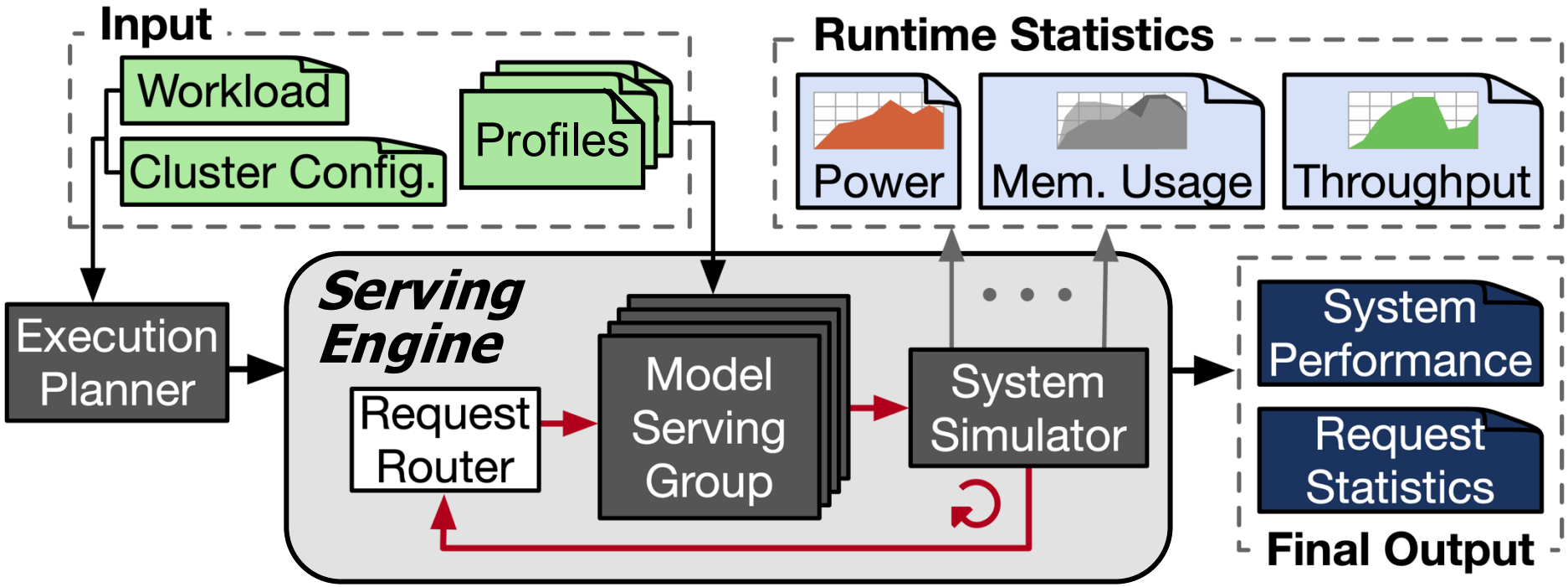
- No support for emerging or heterogeneous hardware
- Coverage remains limited to **a narrow subset**

Toward Unified Simulation of Heterogeneous and Disaggregated LLM Serving

LLMServingSim 2.0

LLMServingSim 2.0 Overview

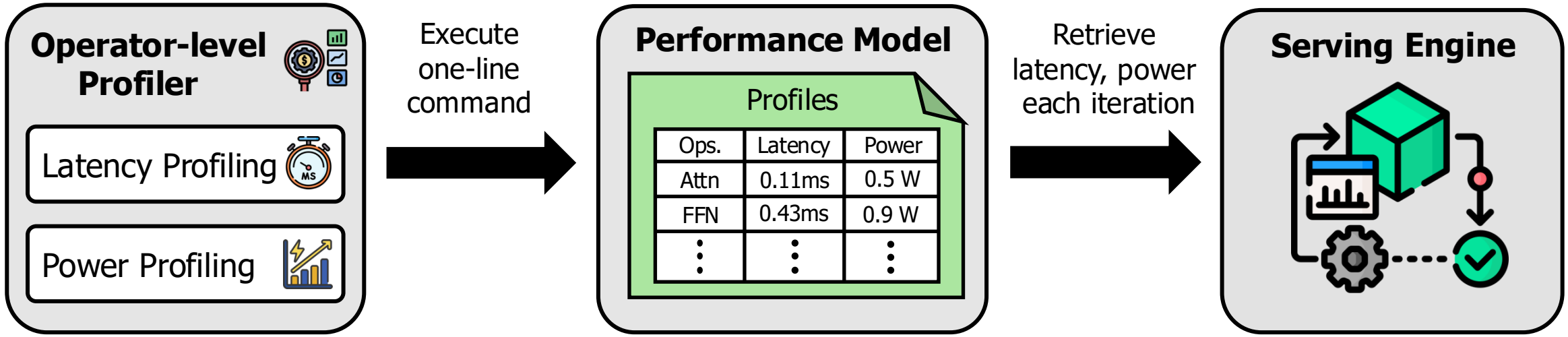
- Simulation from workload traces, cluster configuration, and hardware profiles
- **Iteration-level execution** with model serving groups and system simulator
- **Rich runtime statistics** beyond final summary metrics



Profile-based Performance Modeling

HW Extensibility | Power Modeling | MSG-level Serving | Sys-level Serving

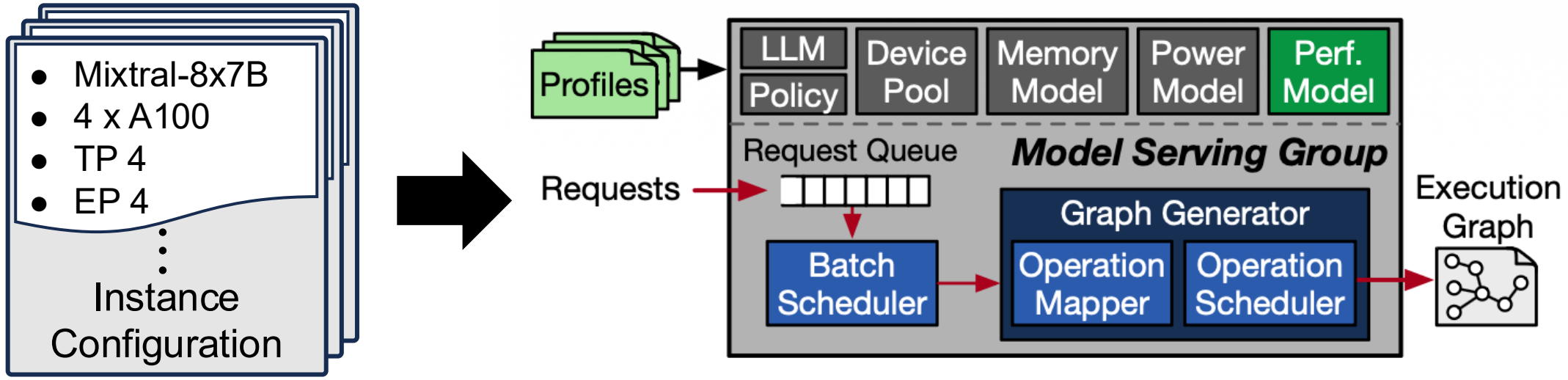
- **Detailed hardware simulation is slow and costly to integrate**
- **Operator-level profiler to build reusable latency and power models**
  - Single-device profiling with minimal code changes via a Pytorch/vLLM-based profiler
  - Reusable profiles for GPUs, TPUs, and future accelerators like PIM



Model Serving Group (MSG)

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

- **An MSG is the logical serving unit for one LLM instance**
- **It encapsulates hardware resources, models, and execution policies**
  - Each MSG is constructed directly from the cluster configuration input



Cluster Configuration **Model Serving Group**

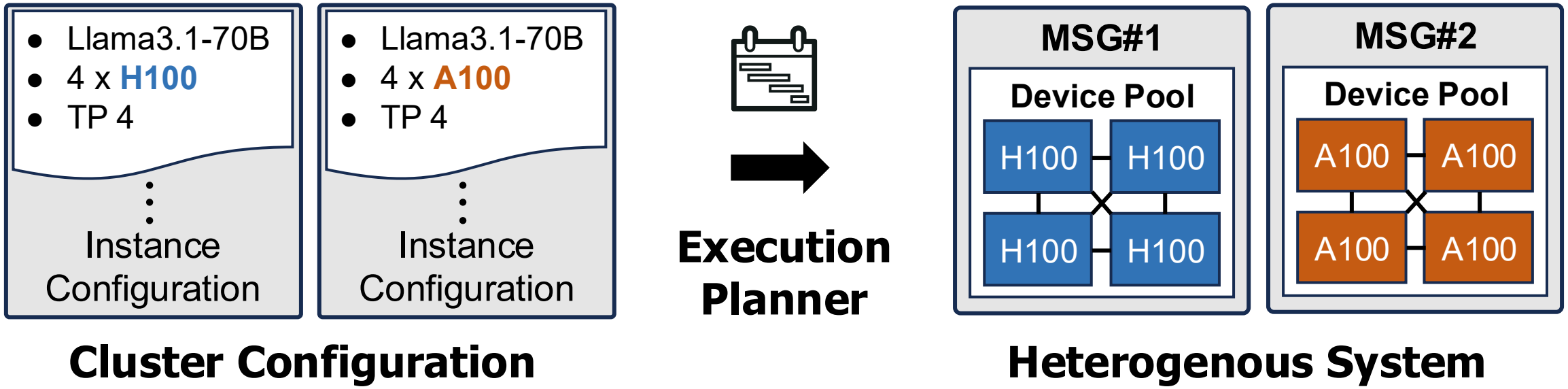
MSG in Action

Making Heterogeneous and Disaggregated LLM Serving Simulatable

Inter-MSG Heterogeneity

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

- **MSG abstraction** naturally enables heterogeneous systems
- **Execution Planner** instantiates MSGs and assigns hardware

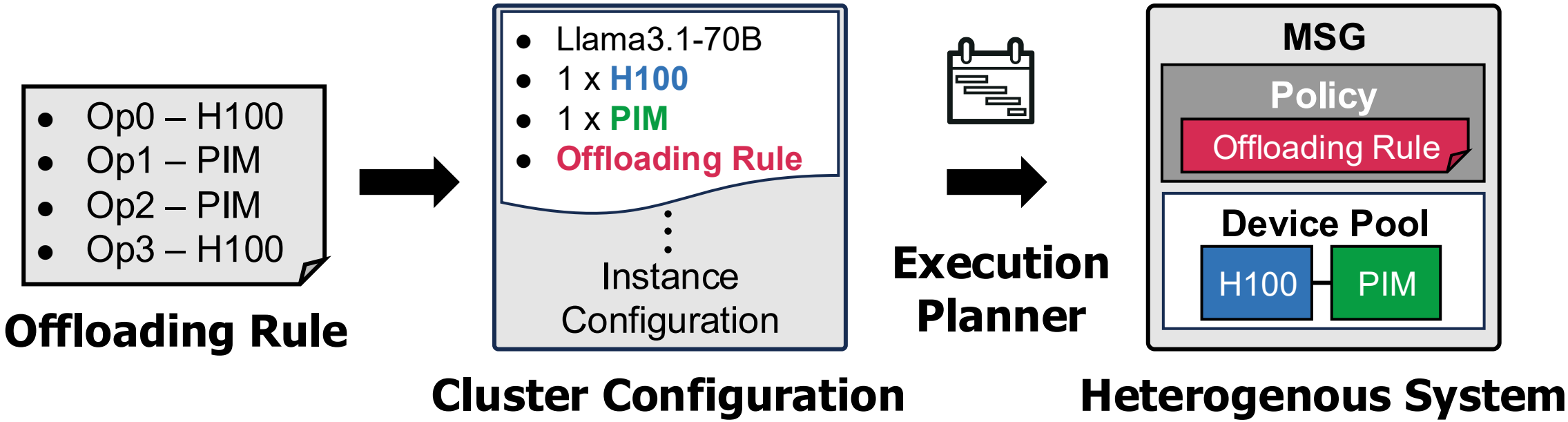


Cluster Configuration **Heterogenous System**

Intra-MSG Heterogeneity

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

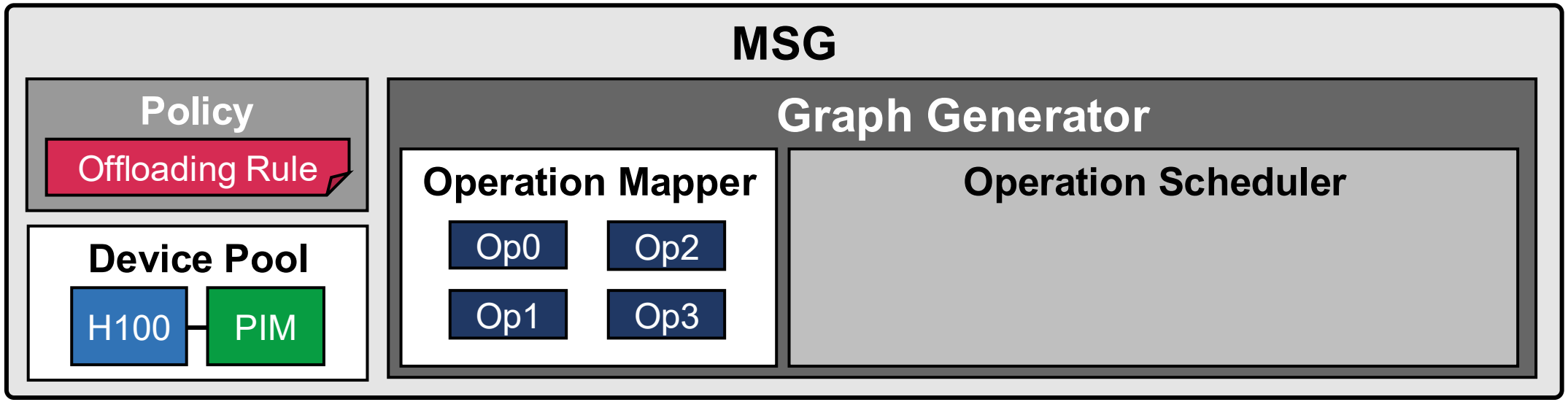
- **Compose heterogeneous devices within a single instance**
- **Specify offloading rules for device-specialized execution**



Handling Heterogeneity in MSG

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

- **Operation Mapper** maps operation to each device with offloading policy
- **Operation Scheduler** adds move and sync operations between devices

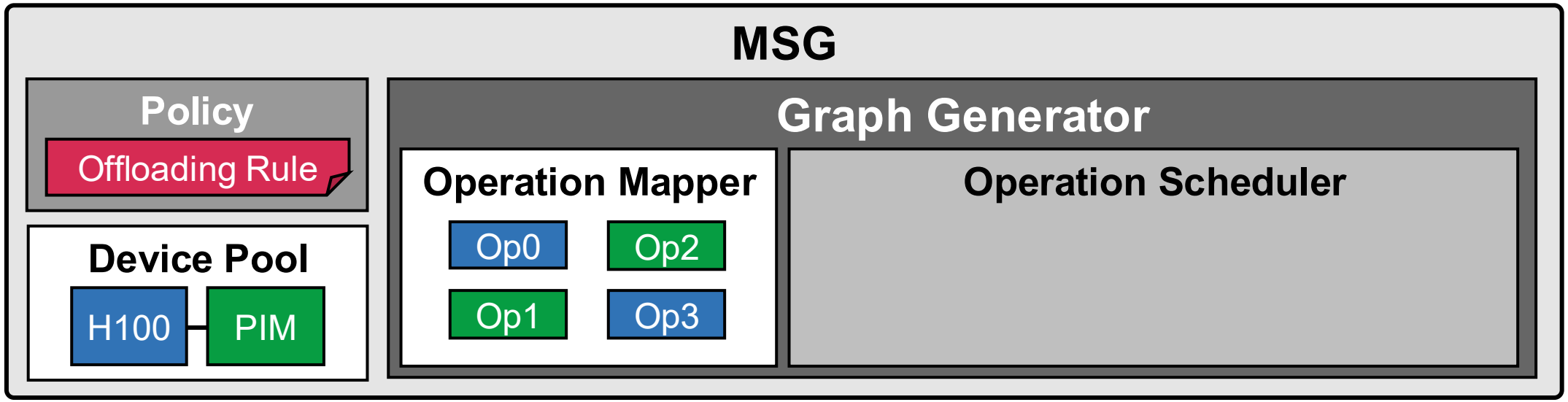


Heterogeneous MSG Workflow

Handling Heterogeneity in MSG

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

- **Operation Mapper** maps operation to each device with offloading policy
- **Operation Scheduler** adds move and sync operations between devices

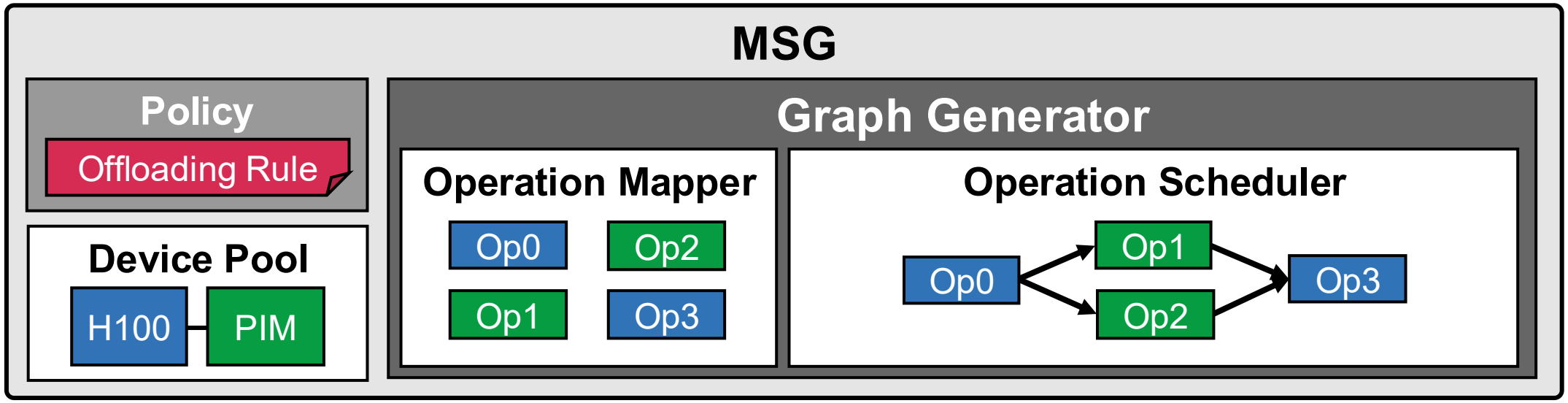


Heterogeneous MSG Workflow

Handling Heterogeneity in MSG

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

- **Operation Mapper** maps operation to each device with offloading policy
- **Operation Scheduler** adds move and sync operations between devices

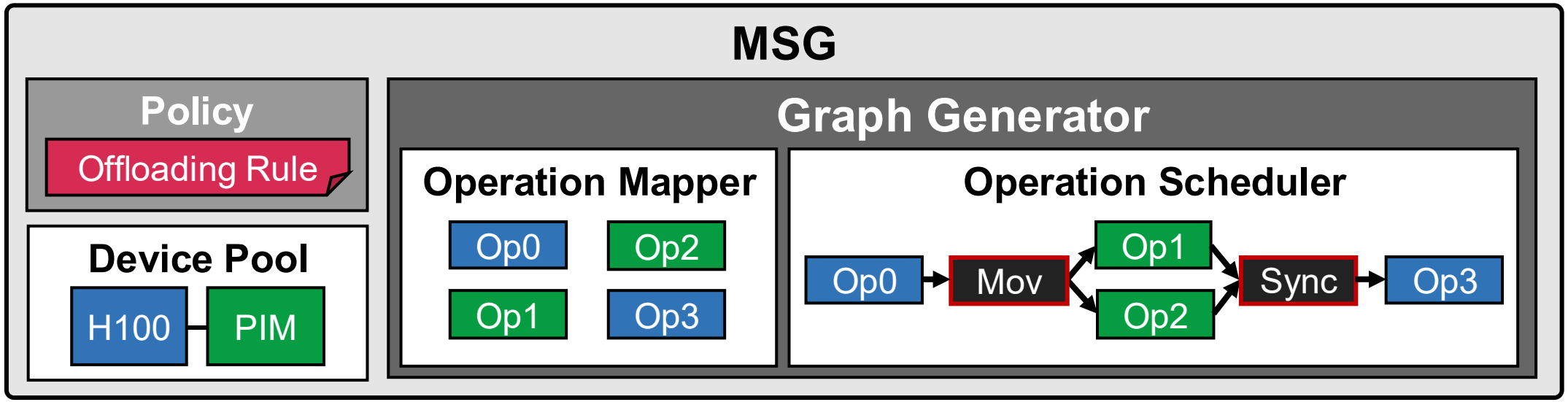


Heterogeneous MSG Workflow

Handling Heterogeneity in MSG

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

- **Operation Mapper** maps operation to each device with offloading policy
- **Operation Scheduler** adds move and sync operations between devices

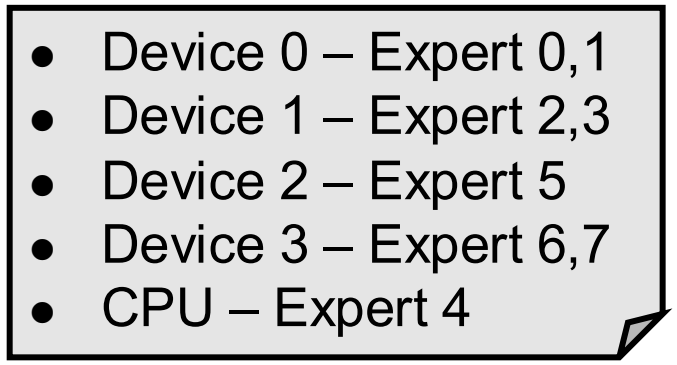


Heterogeneous MSG Workflow

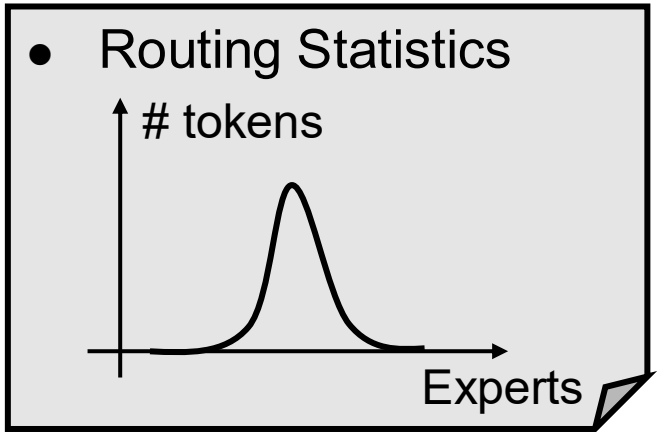
Enabling Mixture of Experts

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

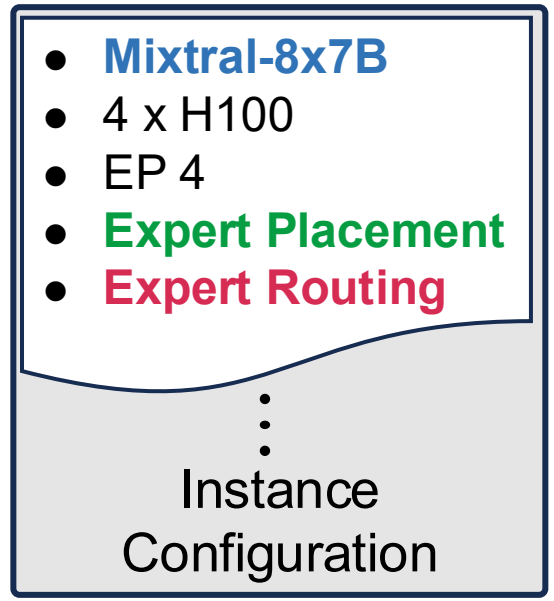
- **Configure instance with MoE model and expert parallelism**
- **Add expert placement and expert routing policy**



Expert Placement



Expert Routing

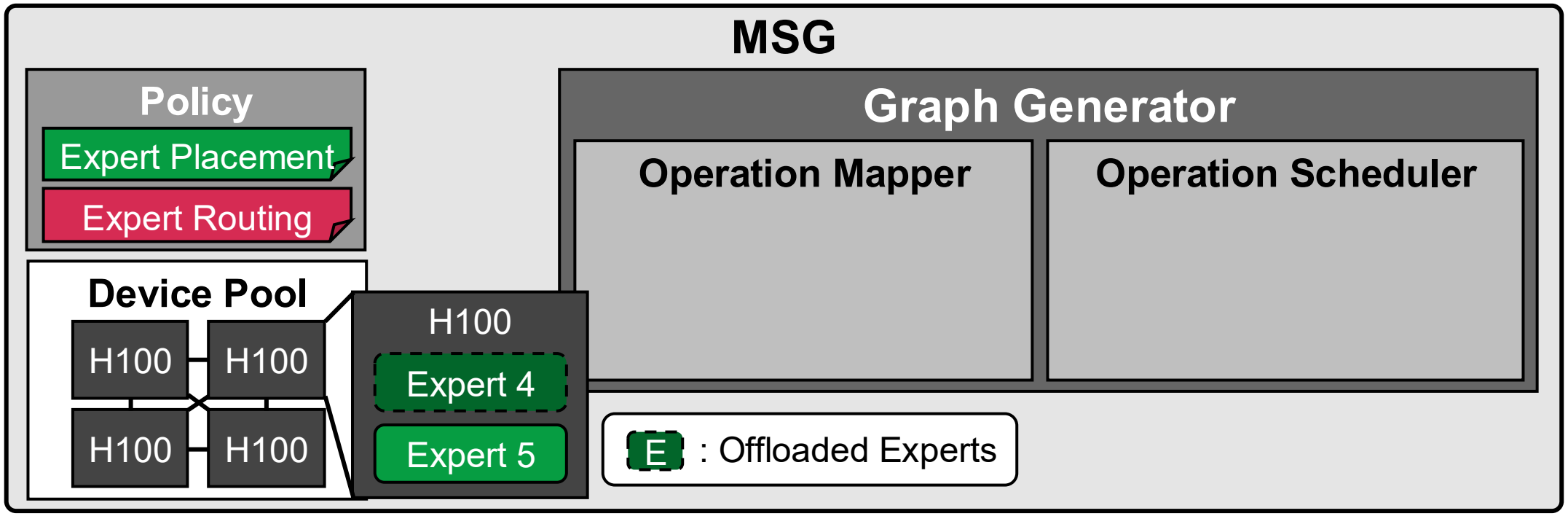


Cluster Configuration

Managing Experts in MSG

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

- **Operation Mapper** routes tokens to experts based on the routing policy
- **Operation Scheduler** loads offloaded experts and inserts sync operations

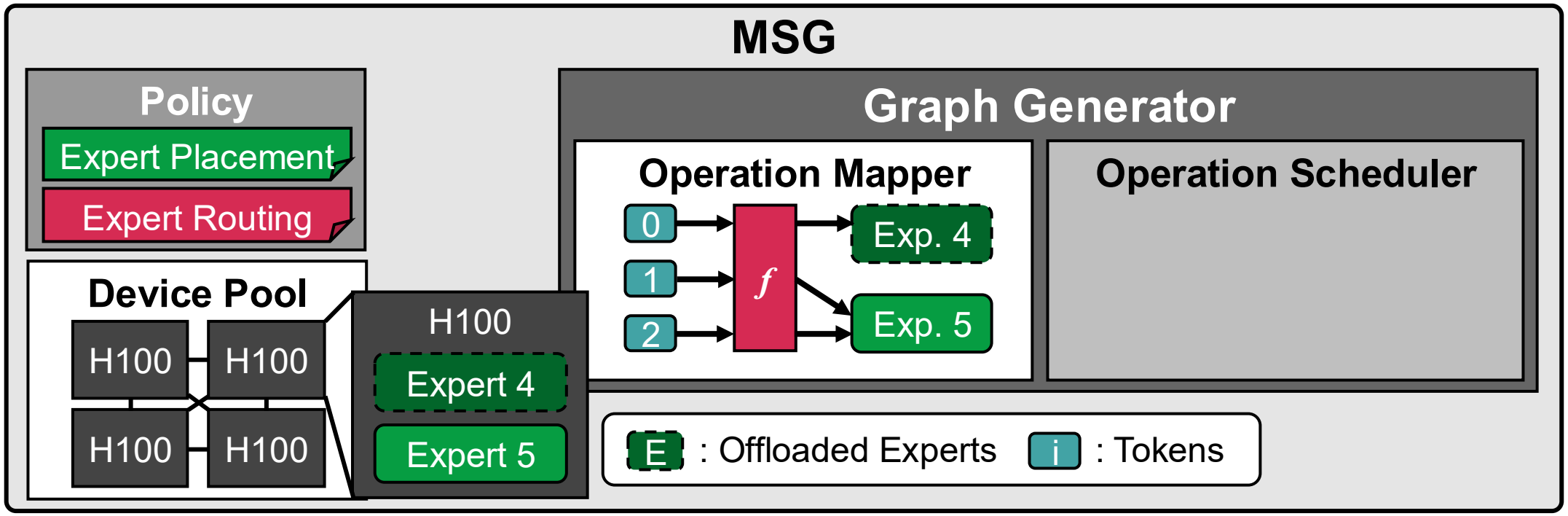


MoE Enabled MSG Workflow

Managing Experts in MSG

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

- **Operation Mapper** routes tokens to experts based on the routing policy
- **Operation Scheduler** loads offloaded experts and inserts sync operations

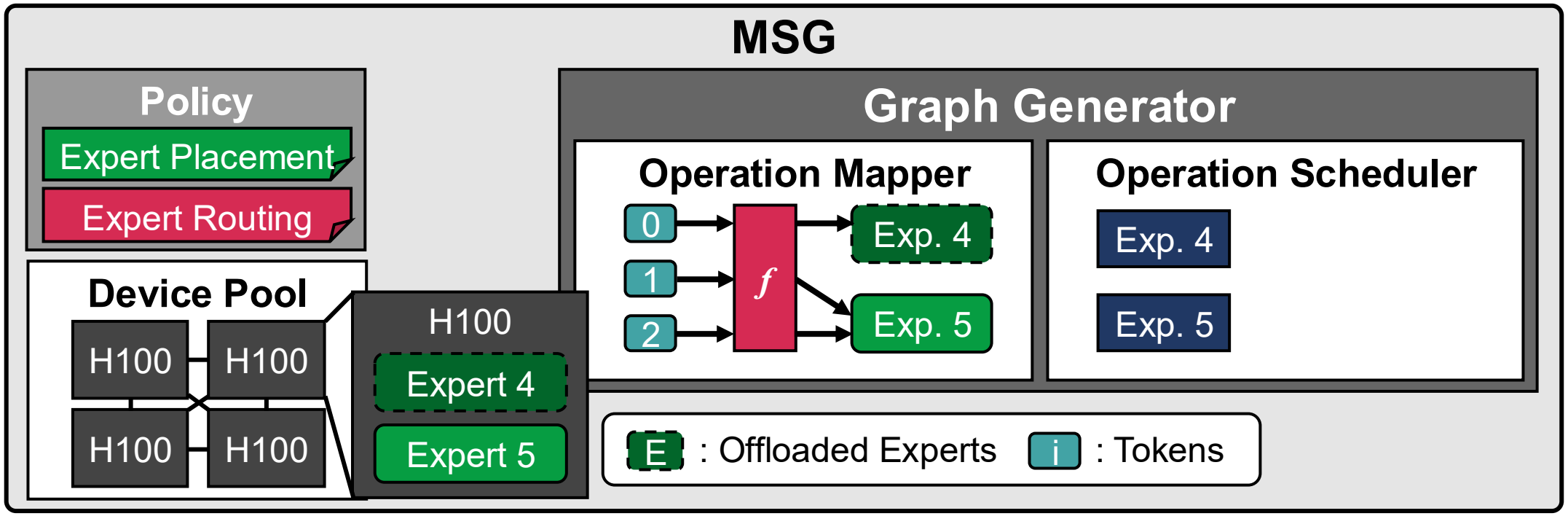


MoE Enabled MSG Workflow

Managing Experts in MSG

HW Extensibility | Power Modeling | **MSG-level Serving** | Sys-level Serving

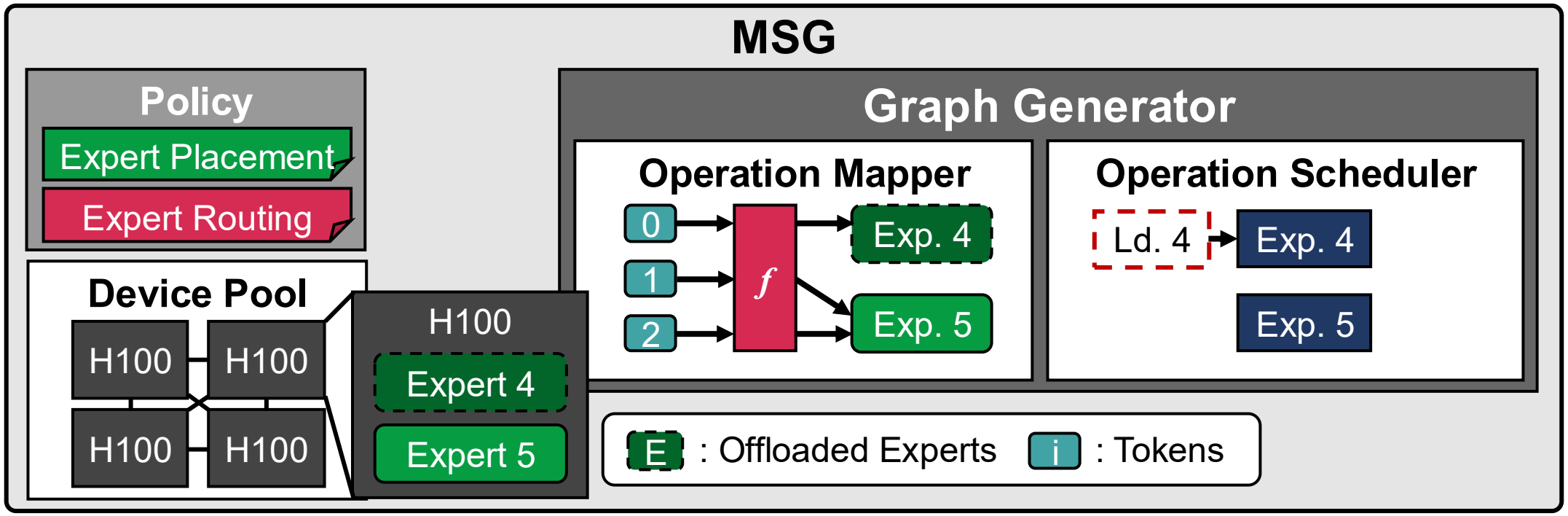
- **Operation Mapper** routes tokens to experts based on the routing policy
- **Operation Scheduler** loads offloaded experts and inserts sync operations



MoE Enabled MSG Workflow

Managing Experts in MSG

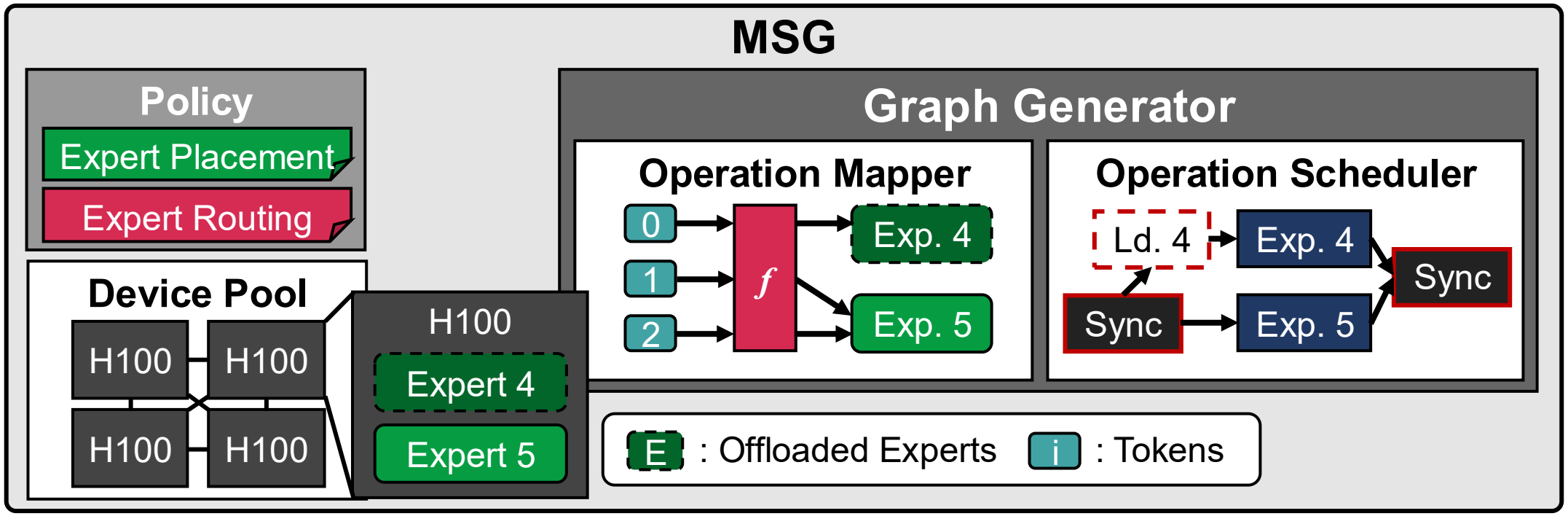
- **Operation Mapper** routes tokens to experts based on the routing policy
- **Operation Scheduler** loads offloaded experts and inserts sync operations



MoE Enabled MSG Workflow

Managing Experts in MSG

- **Operation Mapper** routes tokens to experts based on the routing policy
- **Operation Scheduler** loads offloaded experts and inserts sync operations



MoE Enabled MSG Workflow

Prefill-Decode Disaggregation

HW Extensibility | Power Modeling | MSG-level Serving | **Sys-level Serving**

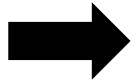
- **Mark each instance configuration as prefill or decode**
- **Configure P/D mapping and KV transfer logic**

- Prefill 0 – Decode 0
- Prefill 1 – Decode 1
- Prefill 2 – Decode 1
- …

P/D Mapping

- token<10: batch
- else:
  layer-by-layer

KV Transfer Logic



- Llama3.1-70B
- 4 x **H100**
- **Prefill**
- **P/D Mapping**
- **KV transfer**

Instance Configuration

M Prefill

- Llama3.1-70B
- 4 x **A100**
- **Decode**

Instance Configuration

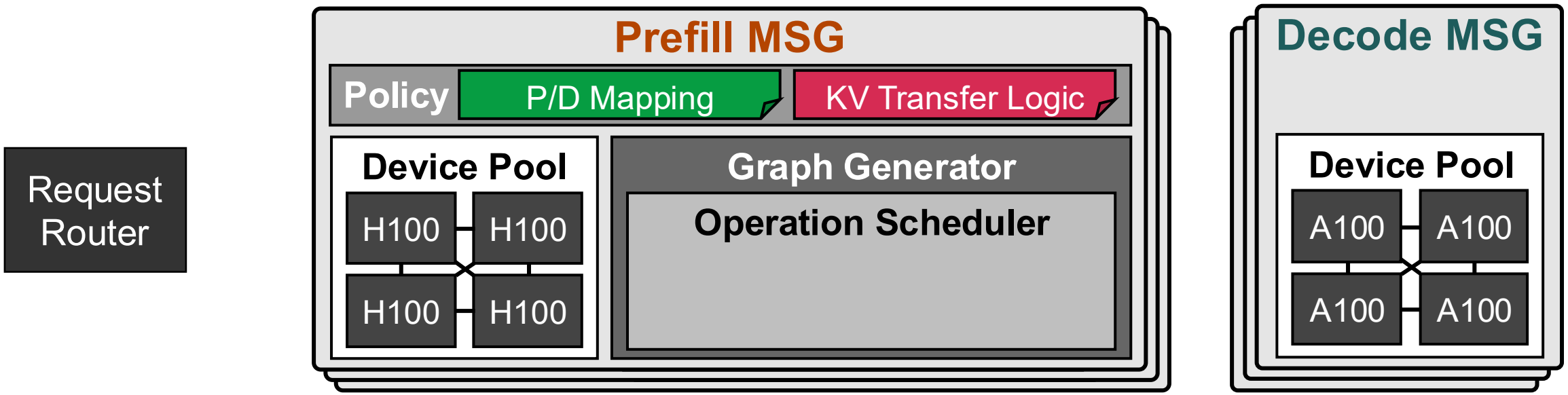
N Decode

Cluster Configuration

Request Routing and KV Transfer

HW Extensibility | Power Modeling | MSG-level Serving | **Sys-level Serving**

- **Request Router** sends requests to each prefill MSG
- Prefill MSG **transfers KV caches** to the corresponding decode MSG

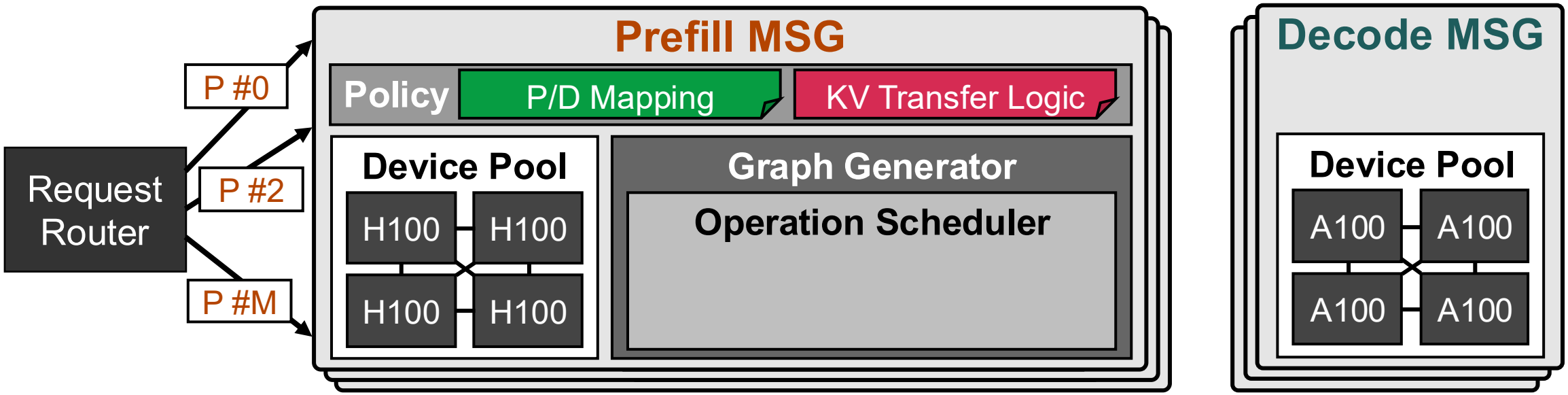


P/D Disaggregated System Workflow

Request Routing and KV Transfer

HW Extensibility | Power Modeling | MSG-level Serving | **Sys-level Serving**

- **Request Router** sends requests to each prefill MSG
- Prefill MSG **transfers KV caches** to the corresponding decode MSG

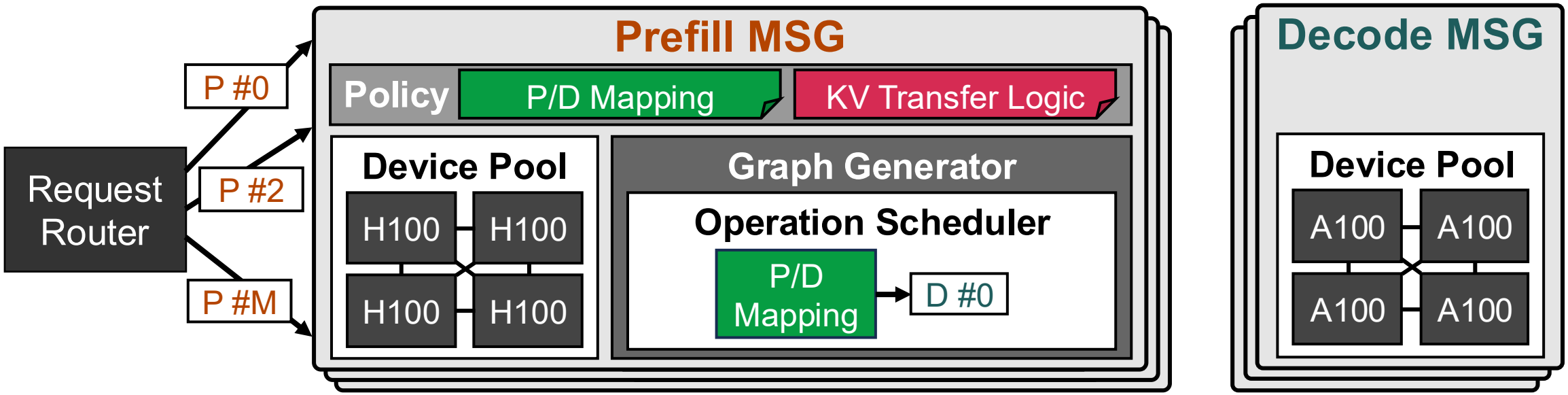


P/D Disaggregated System Workflow

Request Routing and KV Transfer

HW Extensibility | Power Modeling | MSG-level Serving | **Sys-level Serving**

- **Request Router** sends requests to each prefill MSG
- Prefill MSG **transfers KV caches** to the corresponding decode MSG

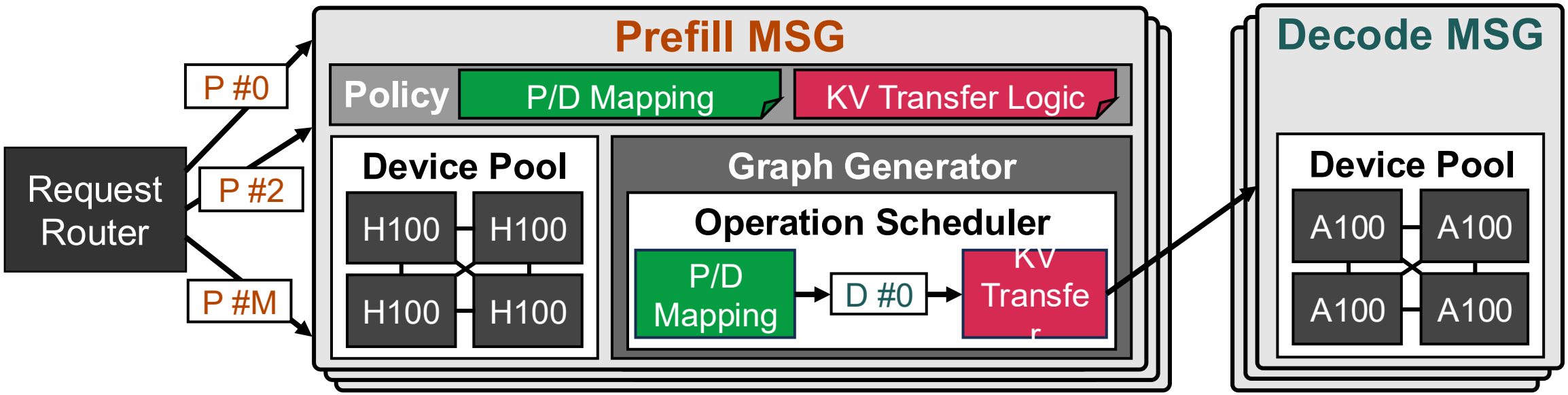


P/D Disaggregated System Workflow

Request Routing and KV Transfer

HW Extensibility | Power Modeling | MSG-level Serving | **Sys-level Serving**

- **Request Router** sends requests to each prefill MSG
- Prefill MSG **transfers KV caches** to the corresponding decode MSG

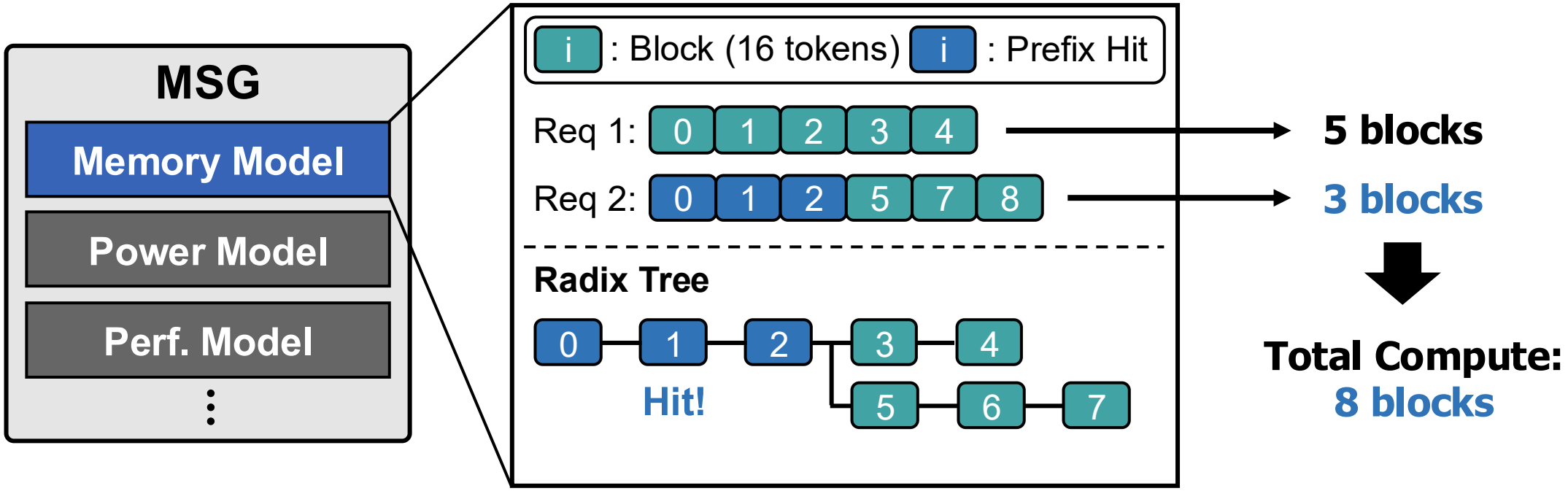


P/D Disaggregated System Workflow

Radix Tree-based Prefix Caching

HW Extensibility | Power Modeling | MSG-level Serving | Sys-level Serving

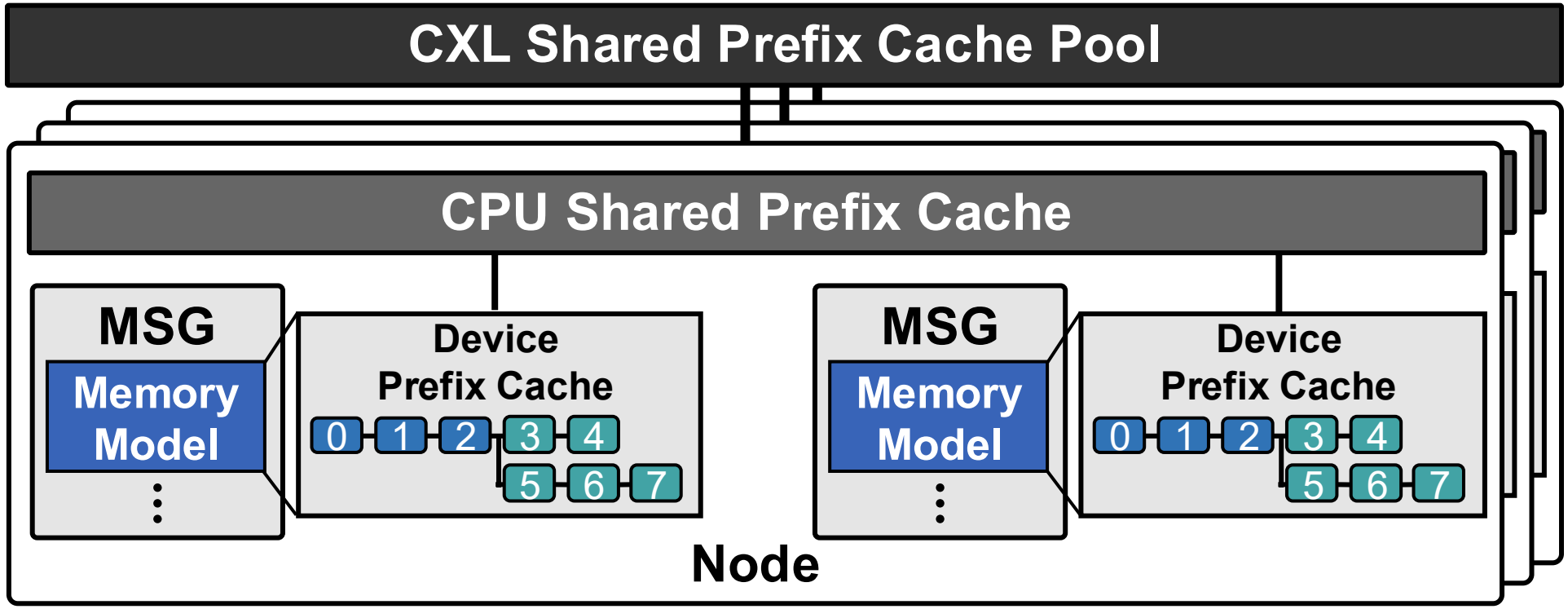
- **Memory model** in MSG manages KV caches in blocks
- Add a **radix tree** to enable prefix matching



Expanding the Memory Model

HW Extensibility | Power Modeling | MSG-level Serving | **Sys-level Serving**

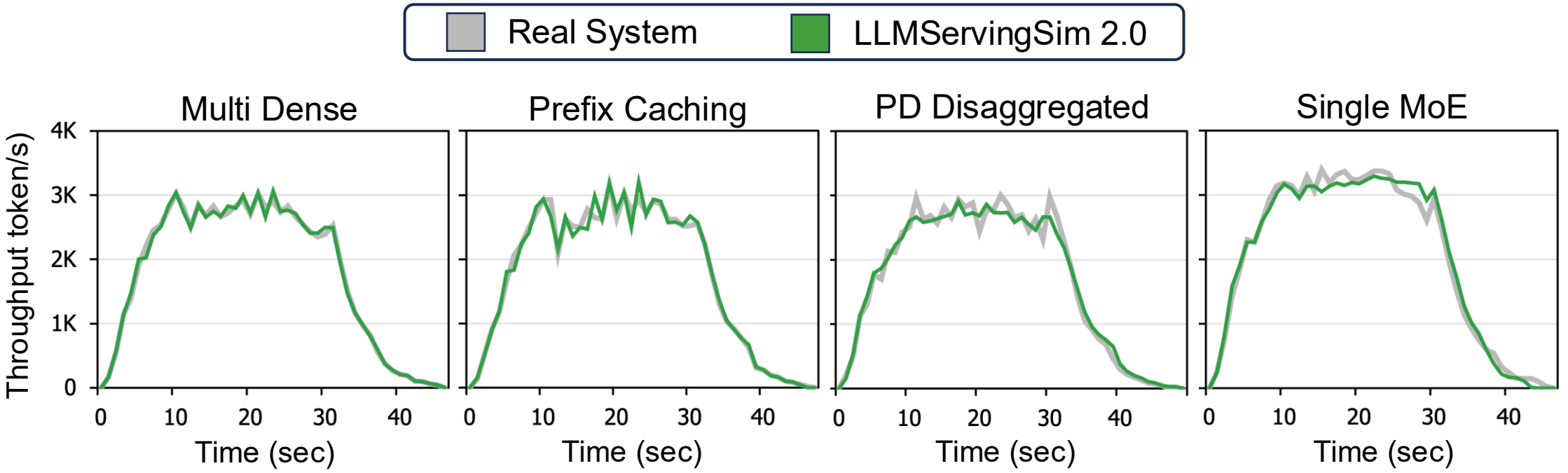
- Add multi-tier memory model (device, CPU, and CXL)
- Allows multiple MSGs to share the prefix cache



Methodology

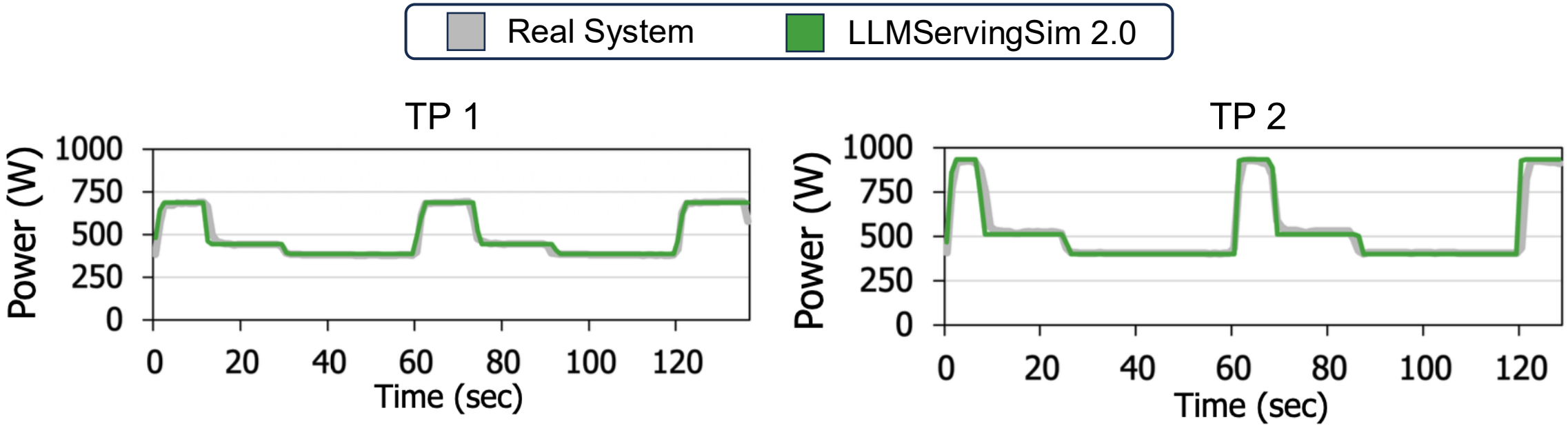
- **Real-Sytem Baseline**
  - vLLM & LMCache
  - 8 x H100 GPUs
  - 4 x A6000 GPUs
  - 1 x TPU-v6e-1
- **Simulator Baseline**
  - LLMServingSim 1.0
  - TokenSim
  - Vidur
  - APEX
- **Models**
  - H100
    - Llama 3.1 70B
    - Mixtral 8x7B
  - A6000/TPU
    - Llama 3.1 8B
    - Phi-mini MoE
- **Dataset**
  - ShareGPT

Validation of LLMServingSim 2.0



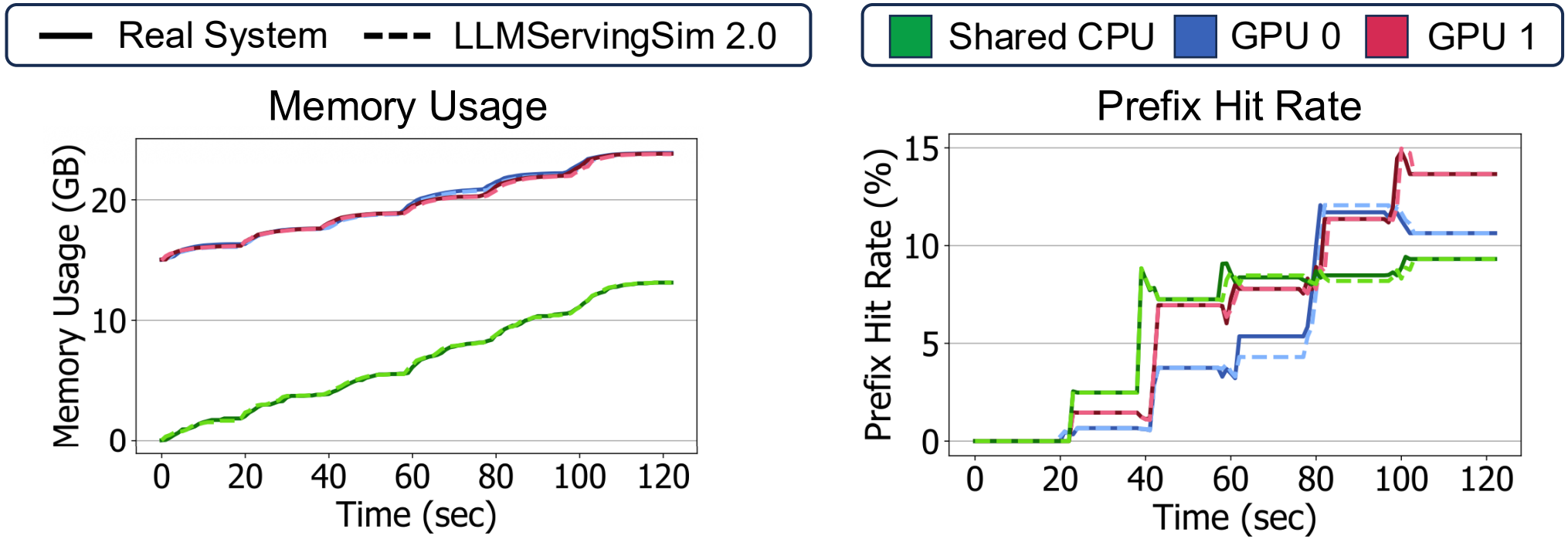
- **LLMServingSim 2.0 closely tracks real-system across all settings**
- **Average point-wise error: 3.29%, Average throughput error: 1.54%**

Validation of Power Modeling



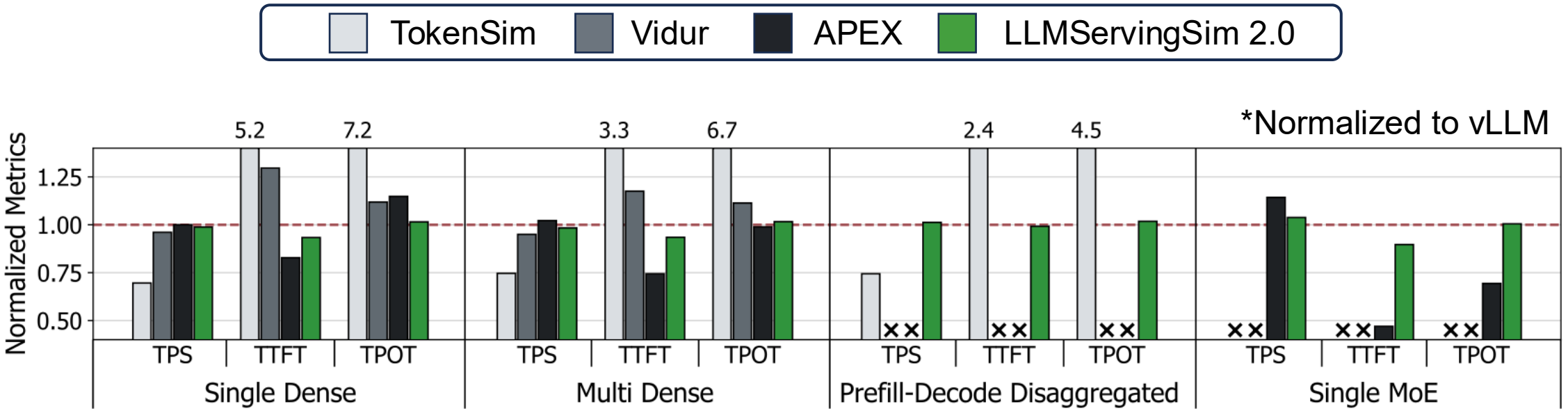
- **LLMServingSim 2.0 closely captures real-system power dynamics**
- **Average error rate 1.34%**

Validation of Memory Modeling



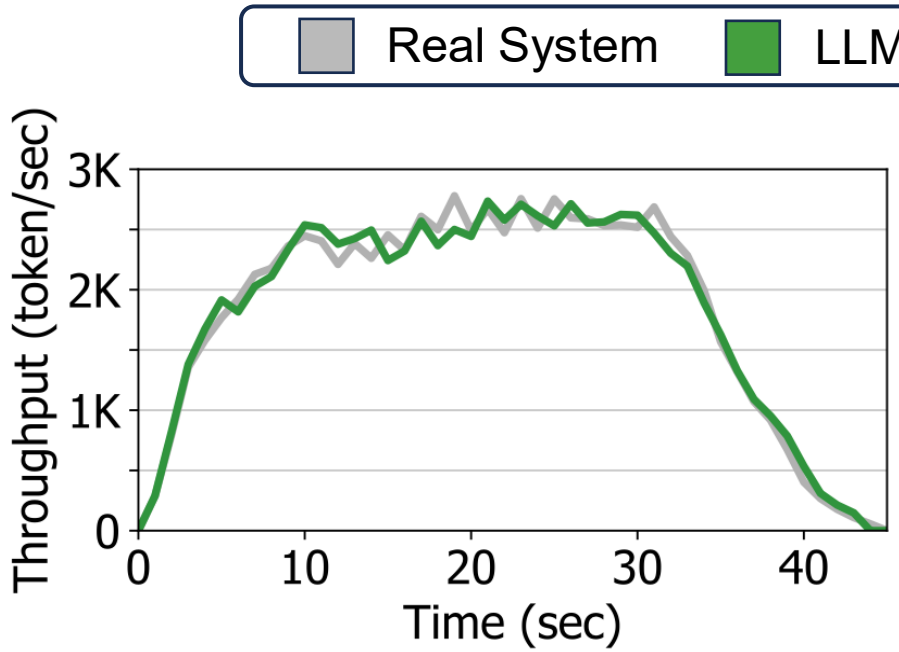
- **LLMServingSim 2.0 closely captures memory and prefix-cache dynamics**
- **Average error rate 0.93%**

Comparison with other Simulators



- **LLMServingSim 2.0 achieves the highest accuracy across diverse scenarios**
- **Average error rate 2.12%**

TPU Case Study



Error Rate (%)	
TPS	0.21795
TPOT	0.23349
ITL	0.50865
Geomean	**0.29581**

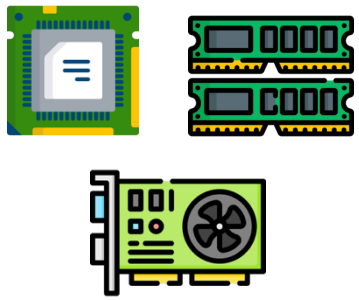
- **LLMServingSim 2.0 closely matches TPU-based real-system behavior**
- **Average error rate 0.3%**

Conclusion

Our simulator code is available
https://github.com/casys-kaist/LLMServingSim

- **LLMServingSim 2.0**
  - A Unified Simulator for Heterogeneous and Disaggregated LLM Serving
- **Contributions**
  - Extensibility to emerging hardware through profile-based modeling
  - Integrated modeling and reporting of power consumption
  - Unified modeling of heterogeneity and Mixture of Experts via MSG abstraction
  - Broad support for emerging system-level serving techniques

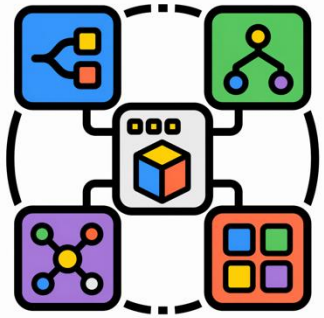
Hardware Extensibility



Power-aware Modeling

MSG-level Serving

Sys-level Serving



Performance

0.95%
Error Rate

<10min
Fast Simulation