MLSys·im Tutorial — Module 4

Design Space Exploration & Synthesis

Vijay Janapa Reddi

Conference Tutorial

Harvard University

Roadmap: Conference Tutorial

Module	Topic	Status
Module 1	Foundations & Architecture	✓Done
Module 2	Advanced Single-Node Analysis	✓Done
Module 3	Scale, Dollars, and Carbon	✓Done
Module 4	**Design Space Exploration & Synthesis**	**← You are here**

TUTORIAL | MODULE 4

Rapid Parametric Sweeps

1

The Combinatorial Explosion

Finding the optimal serving configuration requires testing:

$$|\text{hardware}| \times |\text{batch sizes}| \times |\text{precisions}| \times |\text{parallelism configs}|$$

This space easily exceeds 10^4 configurations. Because `mlsysim` uses analytical math (not cycle-accurate simulation), each evaluation takes $<$ 1 ms.

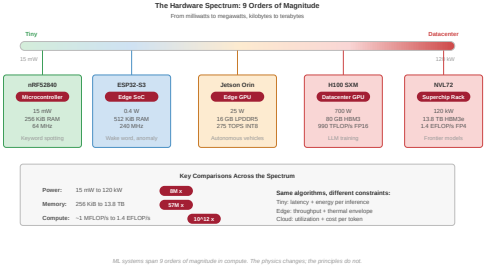
Live Demo: Programmatic Sweeps

```python
from mlsysim.engine.engine import Engine
from mlsysim.hardware.registry import Hardware
from mlsysim.models.registry import Models
import pandas as pd

model = Models.Language.Llama3_8B
hardware = Hardware.Cloud.H100

results = []
for batch_size in [1, 8, 32, 128, 256]:
    perf = Engine.solve(model, hardware, batch_size=batch_size, precision="fp16")
    results.append({
        "Batch Size": batch_size,
        "Throughput (tok/s)": perf.throughput.magnitude,
        "Bottleneck": perf.bottleneck
    })

print(pd.DataFrame(results))
```

TUTORIAL | MODULE 4

TinyML to Frontier

Same Roofline, 9 Orders of Magnitude



Device	FLOPS	TDP
nRF52840	64 M	15 mW
ESP32-S3	500 M	400 mW
H100 SXM	989 T	700 W

Compute Range $\sim 10^{7}\times$

Power Range $\sim 10^{4.7}\times$

The Roofline Model is universal. The physics apply identically to a $2 Microcontroller and a $3M GPU Rack.

TUTORIAL | MODULE 4

Sensitivity Analysis

3

Sensitivity Analysis

```python
from mlsysim.solvers import SensitivitySolver
solver = SensitivitySolver()

result = solver.solve(
    model=Models.Language.Llama3_8B, hardware=Hardware.Cloud.H100,
    precision="fp16", efficiency=0.5)

print(f"Binding Constraint: {result.binding_constraint}")
for param, sensitivity in result.sensitivities.items():
    tag = "<<<" if param == result.binding_constraint else ""
    print(f"  {param:>20}: {sensitivity:+.4f}  {tag}")
```

The Golden Rule: Invest in the parameter with the *largest* partial derivative. Improving a non-binding parameter yields **zero** measurable gain.

TUTORIAL | MODULE 4

SLA-Driven Synthesis

4

Live Demo: Inverting the Roofline

Instead of asking "How fast is this GPU?", what if we ask "What hardware do I need to buy to meet my 30ms latency SLA?"

```python
from mlsysim.solvers import SynthesisSolver
from mlsysim.models.registry import Models
from mlsysim.core.units import Q_

solver = SynthesisSolver()
requirements = solver.solve(
    model=Models.Language.Llama3_8B,
    target_latency=Q_("30 ms"),
    batch_size=1,
    precision="fp16"
)

print(f"Required HBM Bandwidth: {requirements.required_bw.to('GB/s'):.1f}")
print(f"Required Compute:       {requirements.required_flops.to('TFLOPs/s'):.1f}")
```

TUTORIAL | MODULE 4

Capstone & Wrap-Up

Design Challenge: The Capstone

The Problem

$5M budget. Serve Llama-3 70B at **1,000 QPS** with $<$**100 ms TTFT** in **two regions** (US-East + EU-West). Design the fleet.

You must specify using `mlsysim`:

1. **Hardware choice:** Which GPU? How many?
2. **Parallelism strategy:** TP $\times$ PP?
3. **Precision:** FP16? FP8? INT4?
4. **Geographic placement:** Carbon impact?

Resources & Next Steps

Get Started

- `pip install mlsysim`
- GitHub: `harvard-edge/mlsysim`
- Full docs: `mlsysim.readthedocs.io`
- Code cookbook: five interactive scenarios

The Textbook

- *Machine Learning Systems*
- Volume I: Foundations (single node)
- Volume II: Systems at Scale (fleet)
- `mlsysbook.ai`

Key Papers

- Williams et al. (2009)
  Roofline Model
- Chowdhery et al. (2022)
  PaLM / MFU
- Hoffmann et al. (2022)
  Chinchilla Scaling
- Patterson et al. (2021)
  Carbon & Training
- OpenAI (2024)
  o1 / Reasoning Scaling

Use these papers to validate the assumptions behind each solver family.

Thank you! Questions?