Edge Computing

Lecture 04: Edge Systems: Design and Optimization

Recap

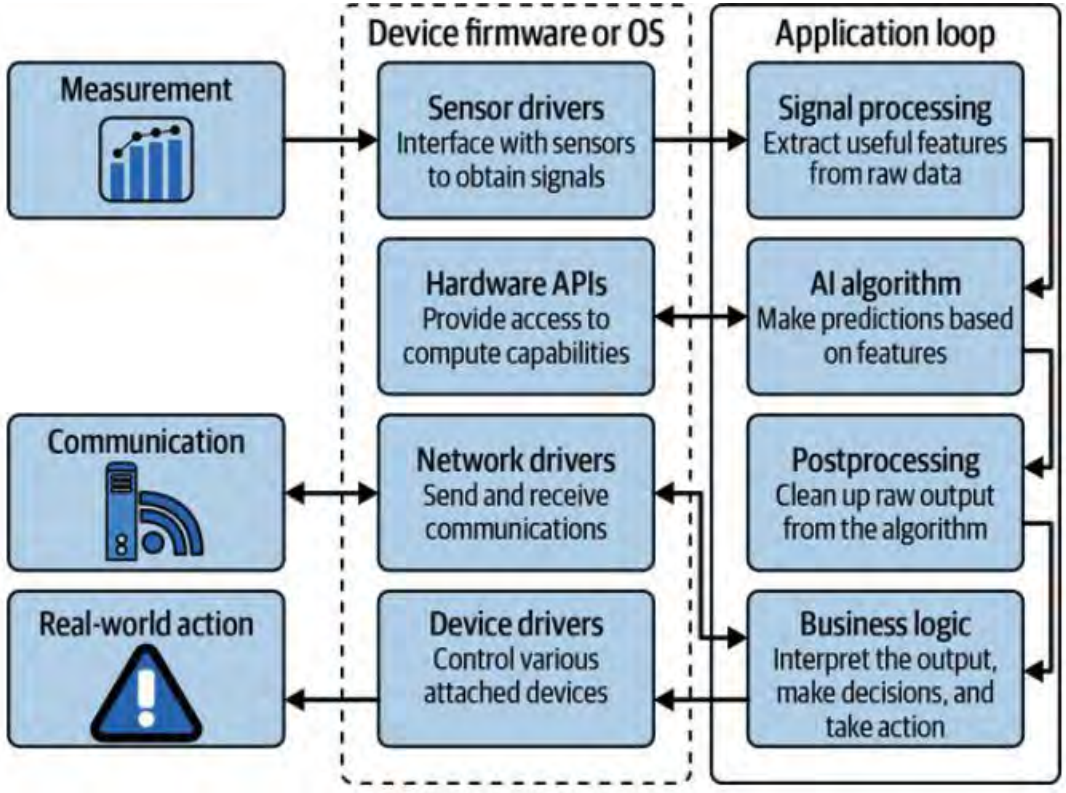
- The IoT Challenge
- Bandwidth, latency, throughput, pipeline
  - Generic metrics, must specify context,
    - e.g. Packet latency vs execution latency/time
    - e.g. Link throughput vs frame throughput (FPS)
- Example system architectures
  - Edge-heavy vs Cloud-heavy apps
- Close-the-loop: sensing, compute, actuation
  - Decision making

Agenda

- Edge System Design and Evaluation
- Edge System Optimization
- Multi-threading & Multi-processing
- Quiz 2

System Design Tips

- Component function
- Logical connection
- Draw a good diagram if possible



Situnayake, D., & Plunkett, J. (2023). AI at the Edge. O'Reilly

System Optimization: Evaluation Metric

- What to optimize?
- What do you care?
- How would you measure?

Metrics

Google it!

System

- FPS
- Throughput
- Latency
- Memory
- Energy
- Thermal
- TOPS/FLOPS

Error

- IoU
- Mean absolute error (MAE)
- Squared MSE/RMSE
- ADE (abs. distance error)

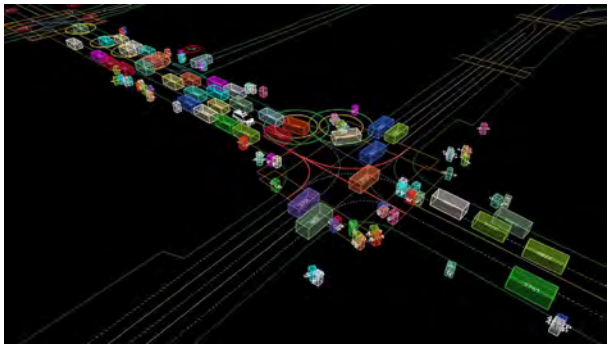
Accuracy

- Loss
- Confusion Matrix

	NO	NOISE	YES
NO	96.3%	0%	3.7%
NOISE	2.7%	95.9%	1.4%
YES	4.7%	0.9%	94.4%

- True/False Positive/Negative
- Precision & Recall
- RUC curve (TP vs FP)
- mAP (mean Average Precision)
  - Area under precision vs recall curve

Detection Task and Evaluation Metric

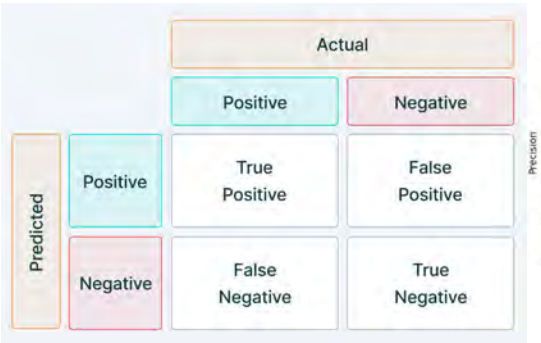


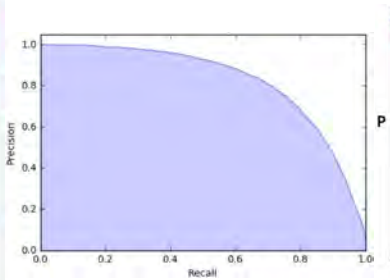
Task

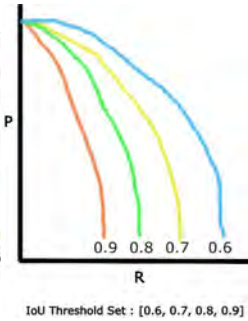
- Produce a set of boxes for the objects in the scene

Metric

- Intersection-over-Union (IoU)
- True/False Positives (TP/FP)
- True/False Negatives (TN/FN)
- Precision/Recall (P/R)
- PR Curve
- Average Precision (AP)







src1, src2

Optimization

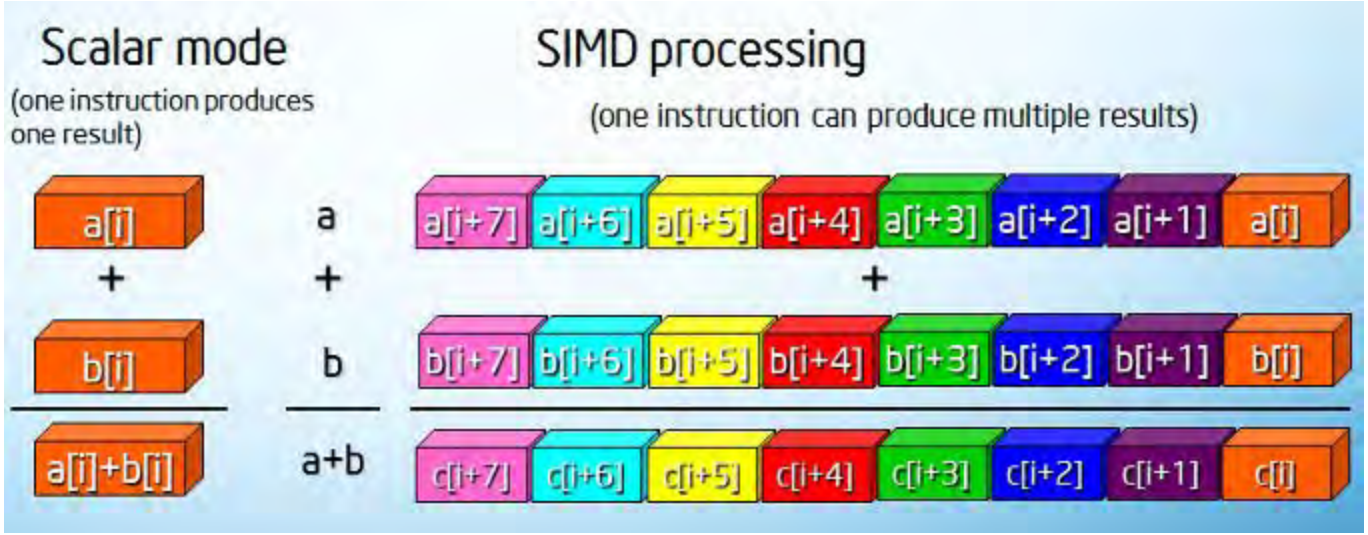
- Design time
  - Topology, device, resource
- Deployment time
  - Evaluate system metrics
  - Workload balancing between edge and cloud
- Run time
  - Evaluate performance metrics at scale
  - Monitoring and supporting

Edge Device Optimization

Facing limits, what can we do?

- Establish baseline (existing technique, or naive implementation)
- Algorithm refinement
- Design tradeoff
- Implementation tailored to hardware

SIMD: Single Instruction/Multiple Data



Concurrency: Thread & Process

Google it!

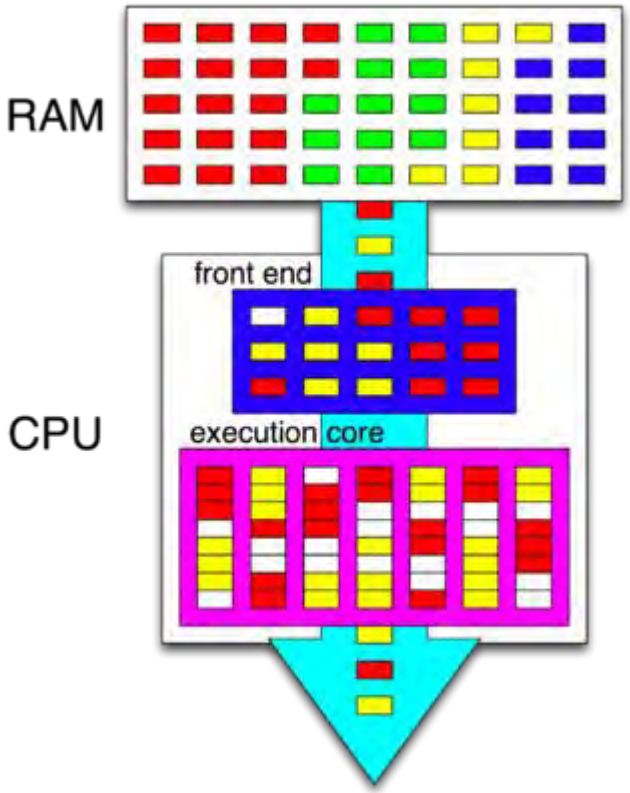
- Thread
  -
- Process
  -

Concurrency: Thread & Process

- A thread is a sequence of instructions within a process. It can be thought of as a lightweight process. Threads share the same memory space.
- A process is an instance of a program running in a computer which can contain one or more threads. A process has its independent memory space.

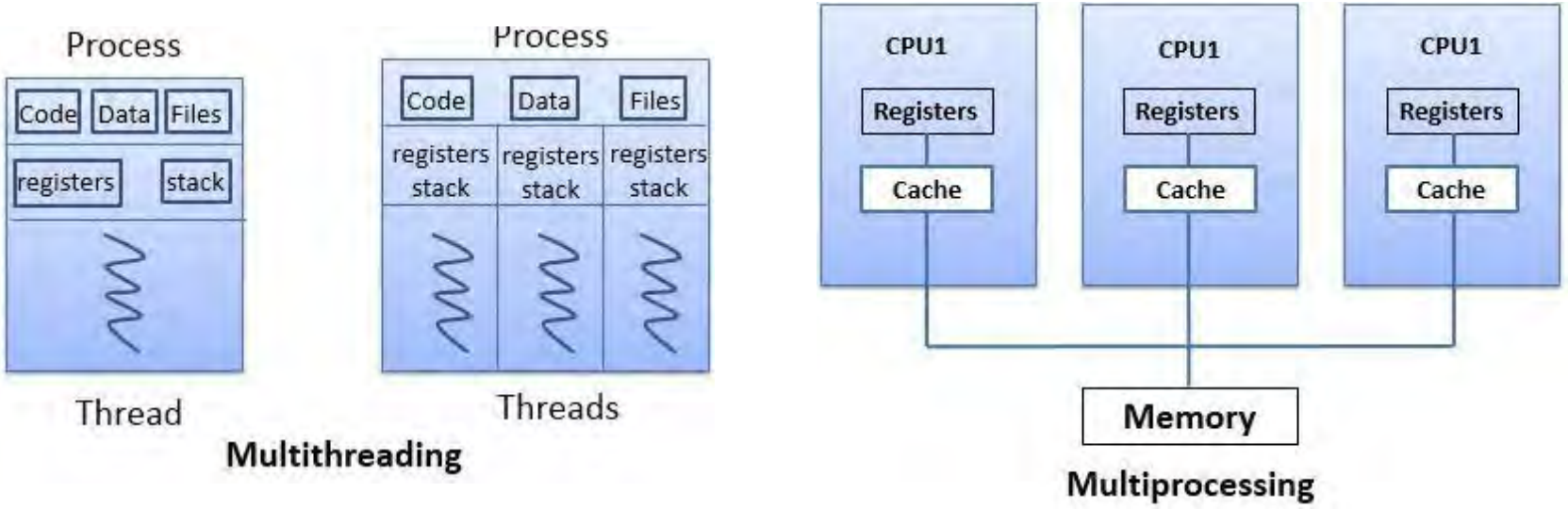
Hyper-threading

- One form of multi-threading
- Each color is a process / thread
- Each box is an instruction
- Frontend blends / reorders the instructions
- Core process instructions from different colors simultaneously
    - White boxes meaning empty resources



https://en.wikipedia.org/wiki/Hyper-threading#:~:text=Hyper%2Dthreading%20(officially%20called%20Hyper,once)%20performed%20on%20x86%20microprocessors.

Multi-Threading & Multiprocess



- Let’s go to Colab: [Profiling performance.ipynb]

Summary

- Edge System Evaluation
  - System Metrics vs Performance Metrics
- Edge System Optimization
  - Architecture design
  - Device optimization
- Concurrency
  - SIMD
  - Threading & Multiprocessing

Next Lecture

- Basics of ML
- Lab 2: object detection
- ML footprint optimization: pruning and quantization